FAMILY GAME COMPUTER

家用电脑学习机

操作手册



北京裕兴电子技术公司

北京裕兴机械电子研究所

集科研、生产、销售、服务为一体

致力于科技开发的裕兴企业

一、1989年率先完成任天堂键盘国产化。

二、1990年率先在游戏机领域申请中国专利，同年推出大型机通用电脑板。

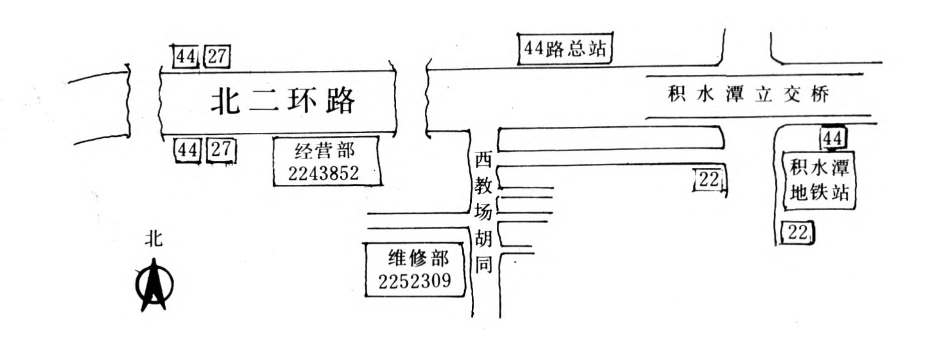
三、1991年率先推出RAM型大容量万用游戏卡。

四、1992年率先推出游戏开发系统，推出中西文游戏机电脑键盘。

五、1993年率先推出基于超级学习卡的游戏机家庭电脑。

六、专利产品超级学习卡集编程、打印、游戏卡、声像卡和可扩展功能为一体，代表当今游戏机领域最高水平。

裕兴企业在游戏机产业拥有7份中国专利，产品种类和水平，雄居首位。



北京裕兴机电研究所

地址：北京西城西教场胡同35号

邮编：100035

电话：2252309　传真：2241471

裕兴普及型电脑

操作手册

总 目 录

汉字输入方法

汉字输入方法目录

汉 字 输 入 方 法

一　让你的名字显示在电视屏幕上

经过一些练习，您应该能顺利地输入各种英文字母、数字和符号了，但对于我们中国人，更需要掌握的是怎样输入汉字，因为我们时刻都在和汉字打交道，操作计算机时也常常要用到汉字，像输入自己的名字、学习英语等。

目前我们所使用的汉字一般都在几千个左右，其中有3755个汉字使用频率最高，它们组成了一级字库，还有3008个不很常用的，组成了二级字库。计算机中一般都有一、二级字库，共6763个汉字。

这么多的汉字，想用只有几十个到一百多个键的键盘直接输入，可不是件容易的事。怎样用英文键盘输入汉字，一度成了世界难题。一大批专家夜以继日地研究，终于解决了这一难题，由此诞生了各种汉字输入方尖，目前使用的较普遍的有全拼、双拼、五笔字型等。裕兴电脑上已安装了“区位”、“全拼”、“双拼”、“五笔字型”等多种输入方法，它们都能输入汉字，各有所长，您可以选一种较喜欢的方法输入汉字。

二　区位输入法

我们先看看区位输入法是怎么回事：

六千多个汉字被分为16－87共71个区，其中每个区又包括0－94共94个位。一个“区位码”由四个数字组成，其中前两个数字表示区，后两个表示位。如：区位码1601，它表示的是第16区的第一位，代表一个汉字“啊”。这样，六千多个汉字就可以用六千多个区位码来表示。

●怎样用区位输入法输入汉字呢？

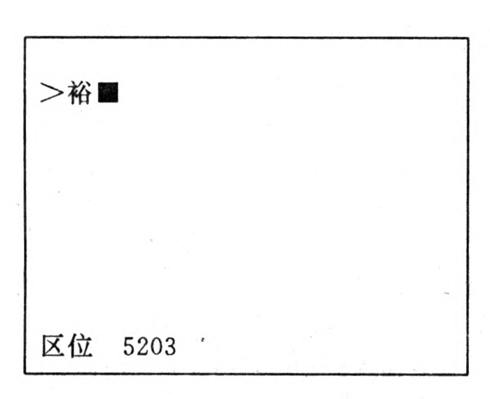
我们以输入汉字“裕”为例，介绍区位输入法的步骤：

①按下“F1”键，屏幕左下角立刻显示出“区位”字样，表示进入了区位状态；

②从《区位表》上找到“裕”字的区位码是“5203”；

③输入这个数字“5203”；屏幕上立刻会显示出一个“裕”字。

④可以按以上步骤继续输入其它汉字，也可以再按一次“F1”键退出区位状态。



区位输入法示意

您现在可以用区位输入法输入您的名字看看（《区位表》在《操作手册》的附录中有，也可

以在书店买到）。

按一下“F1”键，屏幕左下角会立刻显示出“区位”字样，这时侯就可以用区位输入法输入汉字了。

使用区位输入法，需要知道汉字的区位码，如果每输入一个汉字都要查《区位表》，那输入一篇文章所耗费的时间简直就天文数字了！把几千个汉字的区位码都背下来，那是不现实的。因此，区位输入法并不常用。要想更快更有效地输入汉字，就需要掌握更实用的汉字输入方法。

只要您认识汉语拼音，就能使用“拼音”输入法输入汉字。拼音输入法常见的有两种：“全拼”和“双拼”。我们先看看最简单易学的“全拼”输入法：

●我们举例说明全拼输入方法：

以输入汉字“但”为例。

①按一下“F2”键，屏幕左下角显示“全拼”，表示已经进入了全拼状态（还有一些汉字也显示出来，它们一般没有用）；

②我们知道，“但”字的拼音是“dan”，所以我们现在敲入拼音字母“DAN”，敲完后，屏幕最后一行显示：

全拼 DAN■ 1 耽 2 担 3 丹 4 单 5 郸

③五个与“但”同音的汉字都显示出来了，但其中没有我们要输入的“但”字，怎么办呢？我们只要按一下“＞”键，屏幕上立刻会继续显示后面五个同音字：

全拼　DAN■　1 掸 2 胆 3 旦 4 氮 5 但

④其中第五个字是我们要输入的，此时按一下“5”键，“但”字就输入完了，它立刻出现在光标处。

现在可以按上面步骤，继续输入其它汉字，也可以按“F2”键，退出全拼状态。

注：由于汉字中同音字较多，每输入一个拼音，屏幕上每次只显示出五个拼音相同的字，若按“＞”键可以继续下一组拼音相同的字，按“＜”键则可以往回找。

＞但■

全拼　DAN■　1 掸 2 胆 3 旦 4 氮 5 但

全拼输入法示意

●双拼输入法——全拼输入法简单易学，但输入速度却太慢，有一种与全拼类似的输入法，叫双拼输入法，同样简单易学，但速度较全拼有所提高。

双拼输入法和全拼一样，也是输入拼音，得到汉字。但双拼有一个特点：不论拼音多长，一律只敲两个键，其中第一个键是声母，第二个键为韵母。如“F”键，在双拼中，它既可以表示声母“f”，又可以表示韵母“an”。下面仍以“但”字为例，说明双拼输入法：

①按一下“F3”键，屏幕左下角显示“双拼”，表示已经进入了双拼状态（还有一些汉字也显示出来，它们一般没有用）；

②我们知道，“但”字的拼音是“dan”，所以我们现在敲入字母“DF”（“F”可表示韵母“an”），敲完后，屏幕最后一行显示：

双拼　DAN■　1耽2担3丹4单5郸

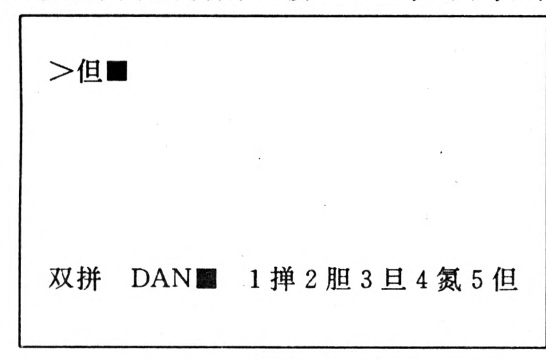
③五个与“但”同音的汉字都显示出来了，其中没有我们要输入的“但”字，按一下“＞”键，继续查找后面的同音字：

双拼　DAN■　1撣2胆3旦4氮5但

④其中第五个字是我们要输入的，此时按一下“5”键，“但”字就输入完了，它立刻出现在光标处。

现在可以按上面步骤，继续输入其它汉字，也可以按“F3”键，退出双拼状态。

注：由于汉字中同音字较多，每输入一个拼音，屏幕上每次只显示出五个同音字，如果没有要输入的，可以按“＞”继续显示其它同音字，按“＜”可以则可以往回找。



双拼输入法示意

我们已经知道字母“F”可以表示声母“f”和韵母“an”，还有许多类似的键，归纳如下，以便记忆：

Q	W	E	R	T	Y	U	I	O	P
ou	ei	r	en	eng	iu	zh	ch		ia

A	S	D	F	G	H	J	K	L	：
	ao	ai	an	ang	ian	iang uang	iao uai	ie	；

Z	X	C	V	B	N	M	<	>	?
un	uan	ui ue	sh uo	ong iong	ing	in	，	.	/

图十三　　双拼键盘排列示意图

您能用上面介绍的输入法输入您的名字了吗？看看哪种输入法速度最快。

●其它汉字输入方法：

裕兴超级学习卡还为你提供了如今社会上很热门的五笔字型汉字输入法。五笔字型汉字输入的详细资料请看《五笔字型练习册》。

五笔字型练习册

目　　录

随着电脑在我国的普及，为适应中国人使用电脑，涌现出很多汉字输入法，其中五笔字型输入法以重码率低，输入速度快等优点得以逐渐被广泛采用。（据统计，经一定的指法训练，五笔字型法输入可达每分钟 120～160 字。）裕兴为满足您的要求，提供了五笔字型输入法，配有五笔字型键帽标签，并以此练习册帮您尽快掌握这种汉字输入法。学习五笔字型既可掌握一种向电脑输入汉字的方法又可以为将来参加社会工作打下基础，还可以学习一些汉字结构方面的知识。您看，何乐而不为呢?!

一 基 本 字 根

人们把汉字中由笔划交叉连接而成的相对不变的结构称为偏旁、部首，而在五笔字型输入法中，把由基本笔划组成的在汉字中结构相对不变的部分称为字根，偏旁、部首并不等价于字根。比如平时常说的木子李，是说李字由“木”和“子”组成，木和子都是五笔的基本字根，所以五笔中称李字由字根“木”和“子”组成。可是平时说的弓长张，虽说是由“弓”、“长”组成，但其中只“弓”是五笔的基本字根，而“长”不是基本字根，还需要继续分解。

五笔字型字根优选的原则是：字根组字的能力强，而且在日常汉语文字中出现次数多（实用频率高）。这些字根可以按较为统一的规则拼形组成汉字，或者说汉字可以按较为统一的规则拆分为基本字根的确定组合，避免产生多种拆分可能，造成一种字根组合可能出现多个字的局面。

经过大量统计和反复试用，五笔字型法选用了 130 个基本字根，请看五笔字型法基本字根总表。由总表可看出，130 个基本字根按起笔的笔划分为五大区，每区内又分五个位，由两位数的区位码来标记，其中十位数为区号，个数位是位号，以 11－55 共计 25 个代码表示。今后凡是提到字根，一律指这 130 个基本字根，也就是说，只有这 130 个基本字根才有资格参加组字编码，其它任何形态的笔划结构，都要理解为是由这 130 个基本字根组成的。因此，这 130 个基本字根既是组字的依据，又是拆字的依据，是对任何汉字及词汇编码的“基本构件”。这 130 个基本字根中又可分为键名字、笔形和基本字根三种，它们都统称为基本字根，后面会一一提到。

二 字根与字根的结构关系

基本字根可以拼合组成所有汉字。在组成汉字时，字根间的位置关系可以分为四种类型——单、散、连、交。

（1） 单 本身就单独成为汉字的字根，这在 130 个基本字根中占很大比重，有八九十个，如王、木、禾、言等。

（2） 散 构成汉字的不止一个字根，而且字根之间保持一定距离，不相连也不相交。如：汉、字、笔、型、培、训等。

（3） 连 五笔字型中字根间的相连关系并非通俗的望文生意的相互连接之意，五笔字型中并不认为以下字是由字根相连得到的，如：足、充、首、左、页、美、易、麦等，而是特指以下两种情况：

①单笔划与某基本字根相连。如：

五笔字型基本字根总表

区	位	代码	字母	基本字根	口诀	高频字
1横起类	1	11	G	王 一五戋	王旁青头戋五一	一
	2	12	F	土士二十 寸雨	土士二干十寸雨	地
	3	13	D	大犬三 古石厂	大犬三(羊)古石厂	在
	4	14	S	木丁西	木丁西	要
	5	15	A	工匚七弋戈廾廿艹	工戈草头右框七	工
2竖起类	1	21	H	目 卜上止	目具上止卜虎皮	上
	2	22	J	日曰 刂 早虫	日早两竖与虫依	是
	3	23	K	口川	口与川,字根稀	中
	4	24	L	田甲囗车皿车力口	田甲方框四车力	国
	5	25	M	山由门贝几	山上贝,下框几	同
3撇起类	1	31	T	禾禾 竹 夂攵	禾竹一撇双人立 反文条头共三一	和
	2	32	R	白手扌 斤斤	白手看头三二斤	的
	3	33	E	月 彡 乃用豕	月彡乃用家衣底	有
	4	34	W	人亻八	人和八,三四里	人
	5	35	Q	金钅勹 儿夕	金勺缺点夕,氏无七	我
4捺起类	1	41	Y	言讠亠 丶广文方圭	言文方广在四一 高头一捺谁人去	主
	2	42	U	立冫 六辛疒门	立辛两点六门疒	产
	3	43	I	水氵小	水旁兴头小倒立	不
	4	44	O	火灬米 业	火业头,四点米	为
	5	45	P	之辶廴宀礻	之宝盖,摘礻(示)衤(衣)	这
5折起类	1	51	N	已己巳乙 尸心忄羽	已半巳满不出己 左框折尸心和羽子	民
	2	52	B	子孑也凵了 耳卩阝	耳了也框向上	了
	3	53	V	女刀九彐臼	女刀媲臼山朝西	发
	4	54	C	又厶巴马	又巴马,丢矢矣	以
	5	55	X	纟幺弓匕	慈母无心弓和匕,幼无力	经

自	丿连目	千	丿连十	产	立连丿
卜	一连卜	不	一连小	主	丶连土

单笔划与基本字根间有明显间距者不能称为相连关系，如：个少么旦幻旧孔乞鱼。

②带点结构，认为是相连的。这类字如：

勺 术 太 主 义 斗 头

这些字中的点与另外的基本字根并不一定相连，其间可连可不连，可稍远可稍近。在五笔字型中把上述两种情况一律视为相连。即不承认它们之间是上下结合或左右结合。这种规定有利于简化、明确地判定汉字的结构。

(4) 交 指两个或多个字根交叉套迭构成的汉字。如：

专	二交乙	申	日交
里	日交土	果	日交木
必	心交丿		

三 汉字分解为字根的拆分原则

上面所说的单，即汉字本身就是一个基本字根，不需要再拆分，这类字的编码有单独规定。

上面所说的散，由于字根之间疏离分立，很容易拆分，您可以自己试着拆一些字。拆分问题集中于要解决连、交及混合型的情况。具体拆分中要注意掌握下面口诀给出的四个要点。

取大优先，兼顾直观，能连不交，能散不连。

以下为拆分实例：

夷：一弓人　　非：三刂三
无：二儿　　重：丿一日土
自：丿目　　生：丿主
天：一大　　于：一十
丑：乙土

取大优先也叫能大不小。在可能的拆分中以拆分出的字根数量最少的那种拆法为优先。要字根数少且字根尽可能大来实现拆分。尽可能大，指某一拆分出的字根再加一笔便不能构成已知字根为准。请看以下实例：

	正确	错误	
果	日木	旦小	(因为旦不是基本字根)

相连关系，按上面的规定，只是单笔划与基本字根之间的关系才视为连。这类字也就直接拆分为单笔和基本字根两者的组合。这类字如：

不	一小	主	丶王	太	大丶
产	立丿	生	丿主	于	一十
下	一卜	术	木丶	自	丿目
斗	冫十	正	一止	开	一廾
天	一大	头	冫大	入	丿丶
习	乙 冫	灭	一火	血	丿皿
升	丿廾	壬	丿士	刃	刀丶
勺	勹丶	舌	丿古	玉	王丶
叉	又丶	歹	一夕	刁	乙一
乏	丿之	凡	几丶	户	丶尸
亡	亠乙	西	西	夭	丿大
兀	一儿				

拆分中还应注意，一个笔划不能割断用在两个字根中。如：

	正确	错误	
果	日木	田木	（因为写“果”字时中间那竖是一贯而下的）

因此刚才那个口诀不妨加上四句，补为下面这样：

单勿需拆　散拆简单　难在交连　笔划勿断

能散不连　兼顾直观　能连不交　取大优先

四　汉字的三种结构

成千上万的方块汉字大体可分为三种结构类型：左右型、上下型、杂合型。这三种字型的划分是基于对汉字整体轮廓的认识，指的是整个汉字中字根之间的相互位置关系，搞清这一点对于确定多字根的汉字的类型是十分重要的。

1. 左右型汉字

在左右型汉字中，包括两种情况：

（1）　在双合字中，两个部分分列左右，整个汉字中有着明显的界线，如：肚、胡、理、胆、咽、拥等。

咽右边的因由两个字根构成，虽然这两个字根之间是一外一内的关系，但整个汉字却属于左右型。

（2） 三合字中，整个汉字的三个部分从左到右并列，或者单独占据一边的一部分与另外两个部分呈左右排列，如：侧、别、谈等，都应属于左右型。

2. 上下型汉字

上下型汉字也包括两种情况：

（1） 在双合字中，两个部分分列上下，其间有一定距离，如：字、节、看等。

（2） 三合字中，三个部分上下排列，或者单占一层的部分与另外两部分作上下排列，如：意、想、花等。

3. 杂合型——外内型和单体型汉字

杂合型指组成整字的各部分之间没有简单明了的左右上下型关系。如：团、同、这、斗、头、飞、本、天、册、成等。

汉字的三种字型结构

字型代号	字型	字　　例
1	左右	汉　相　结　到　动
2	上下	字　室　花　型　早
3	杂合	困　凶　这　司　乘

汉字的图形特征，是每一个有文化的中国人从上小学起就熟知的。这里，可以用来作为识别汉字的一个重要依据。如："口"、"八"上下排列为"只"，而左右排列为"叭"等。因此，还可以把三种字型叫做字根的三种排列方式。在向计算机输入汉字时，除了键入组成汉字的字根外，有时还有必要告诉机器那些键入的字根是以什么方式排列的，即补充键入一个字型信息，后面会提到。

各型的划分中，还有以下约定：

凡属字根相连（指单笔与字根相连或带点结构）一律视为杂合型。

凡键面字（本身就是单个字根），有单独编码方法，不必利用字型信息。

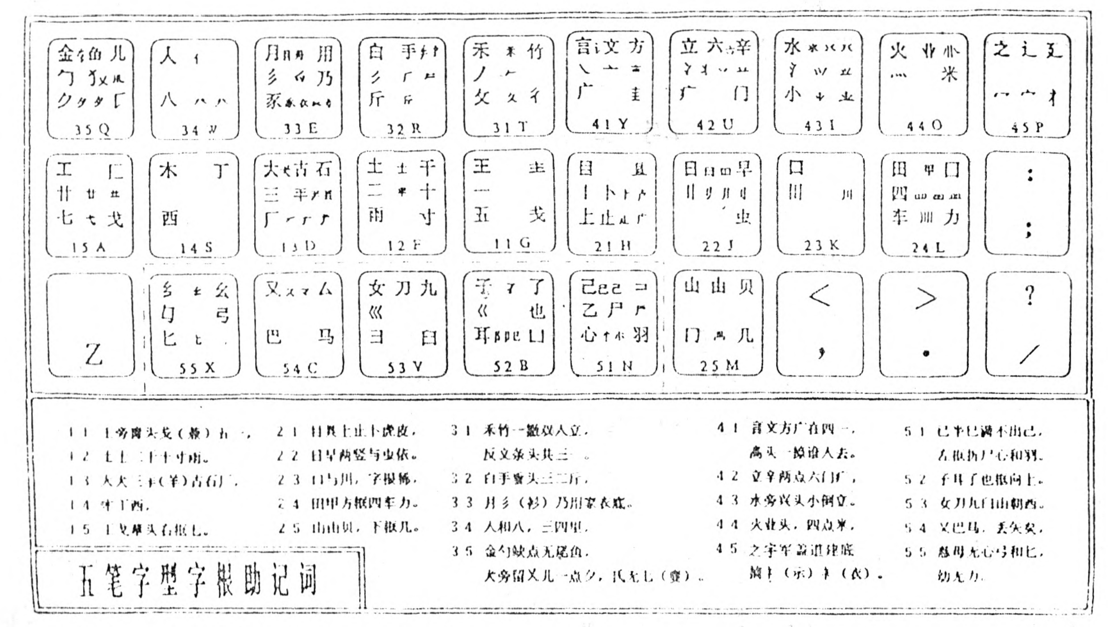
主要对属于散、交两类字根结合关系，要区分字型。

五　五笔字型键盘分布及使用

1　五笔字型字根的键盘分布

五笔字型法的130个字根，按起笔笔划分五大区，每一区占键盘上相连的一片每一片共有五个键位，键位上的编号称为位号。这些位号与字根的关系都反映在字根总表上。按这个表得到的键盘分布如下图：如果您的键盘上没有印下图，可以购买裕兴为您提供的五笔字型键帽标签贴在键上，等您练到熟练的程度就可以揭下键帽盲打了。

五笔字型键盘字根总图



每个键上取一个字根做键名，名谱如下：

一区：	横起笔，	王土大木工	二区	竖起笔，	目日口田山
三区：	撇起笔，	禾白月人金	四区：	捺起笔，	言立水火之
五区：	折起笔，	已子女又纟			

2　键位安排中一些辅助记忆的特点

键位安排上力求有规律、不杂乱，尽量使同一键上的字根在形、音、义方面能产生所需的联想，这有助于记忆，便于迅速熟练掌握。可以列出以下的规律性：

(1)　字根首笔笔划代号和所在区号一致。

(2)　相当一部分字根其第二笔笔划号与位号一致，如：王、丶、戈、文、方、广。

(3)　部分字根的笔划数与位号一致，如：丶、冫、氵、灬分别在 1、2、3、4 位，字根一、二、三分别在 1、2、3 位。

(4)　部分字根与键名字根形态相近，例如：

键　名	形似字根
王	五
土	士、干
大	犬
田	甲、四
山	由
禾	禾
月	月、用
水	氵、水、小
之	辶、廴
已	巳、己、已、尸

（5） 位号从中间向两侧由小到大规则变化。在五笔字型键盘字根总图上逐个键位给出了助记解说，利用这些助记特性和口诀，不难记忆各字根的键位。您不妨仔细观察分析一下字根总表和键盘图，寻找、记住可助记的各类特征。

六 五笔字型单字输入编码规则

1 编码歌诀

单字的五笔字型输入编码有歌诀如下：

五笔字型均直观，依照笔顺把码编；
键名汉字打四下，基本字根请照搬；
一二三末取四码，顺序拆分大优先；
不足四码要注意，交叉识别补后边。

歌诀中包括了以下原则：

（1） 取码顺序，依照从左到右，从上到下，从外到内的书写顺序（见“依照笔顺把码编”那一句）

（2） 键名汉字（见“键名汉字打四下”那一句）

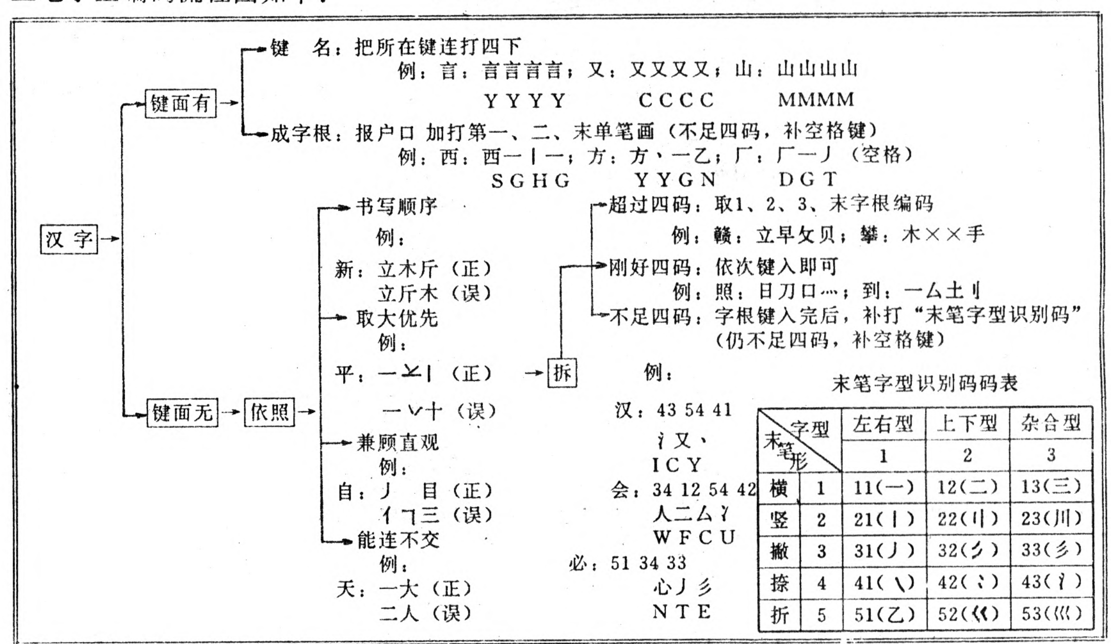
（3） 字根数为四或大于四时，按一、二、三、末字根顺序取四码（见“一二三末取四码”那一句）

（4） 不足四个字根时打完字根识别码后，补交叉识别码于尾部。此种情况下，码长为 3 或 4（见最后一句）

歌诀中“基本字根请照搬”句和“顺序拆分大优先”是拆分原则。就是说在拆分中以基本字根为单位，并且在拆分时“取大优先”，尽可能先拆出笔划最多的字根。或者说拆分出的字根数要尽量少。

五笔字型编码流程图如下：

五笔字型编码流程图



末笔字型识别码码表

末笔形 \ 字型		左右型	上下型	杂合型
		1	2	3
横	1	11(一)	12(二)	13(三)
竖	2	21(丨)	22(刂)	23(川)
撇	3	31(丿)	32(彡)	33(彡)
捺	4	41(㇏)	42(冫)	43(氵)
折	5	51(乙)	52(巜)	53(巛)

七　键名汉字的编码

我们现在说说键名字，键盘上共有 25 个键名汉字，即：

王　土　大　木　工

王　土　大　木　工

目　日　口　田　山

禾　白　月　人　金

已　子　女　又　纟

这 25 个字各占一键，它们的编码是把所在键的字母连写四次，输入它们时需连击所在键四次，即：

“王”字编码为：GGGG，输入时需连击 G 四下。

“目”字编码为：HHHH，输入时需连击 G 四下，等等。

所以这样规定，是由于已把这些单键分给 25 个高频字，对 25 个高频字击一下便可输入一个汉字，而键名只好委屈些和其它字统一使用四码，25 个高频字的输入见一级简码。

在 130 个基本字根中，除 25 个键名字根外，还有几十个本身也是汉字，称它们为“成字字根”。键名和成字字根合称为键面字。成字字根的编码公式为：

键名码＋首笔码＋次笔码＋末笔码

当成字字根仅为两笔时，只有三码，公式为：

键名码＋首笔码＋末笔码

键名码即所在键字母，击此键又称报户口。

首单笔码、次单笔码和末单笔码，不是按字根取码，而是按单笔划取码，横、竖、撇、捺、折这五种单笔的单笔划取码就是各区的第一个字母，对应关系如下：

笔　划	横	竖	撇	捺	折
单笔划码	G	H	T	Y	N

下面给出几个成字字根的编码：

五：G　G　H　G

雨：F　G　H　Y

木：S　G　H　Y

二：F　G　G

丁：S　G　H

单笔划横和汉字数码“一”以及汉字“乙”（单笔划折的代表）都是只有一笔的成字字根。用上述公式不能概括，而单笔划有时也需要单独使用，特别规定了五个笔划的编码如下：

一：G　G　L　L

丨：H　H　L　L

丿：T　T　L　L

丶：Y　Y　L　L

乙：N　N　L　L

编码的前两位可视为和前述公式有统一性，第一位是键名码，第二位是首笔划码，因为没有

其它笔划而补打两次 L 键。

八 键外字的编码

上述二、三中的键面字总共有一百多个。键面字以外的汉字都是键外字，而且数量很大。汉字输入编码主要是键外字的编码，含四个或四个以上字根的汉字，用四个字根码组成编码，不足四个字根的键外字要补打一个字型识别码。

1. 字根码

每个字根都分派在一个字母键上，其所在键上的英文字母就是该字根的“字根码”。

凡含四个或四个以上字根的汉字，取其第一、二、三、末四个字根码组成键外字的输入码。这里一、二、三、末四个字根应按正常书写顺序，**先左后右，先上后下，先外后内**。下面给出一些例字：

	字根分解	编　码
续	纟十乙大	XFND
紧	刂又纟小	J C X I
容	宀八人口	PWWK
磨	广木木石	YSSD
酸	西一厶夂	SGCT

2. 末笔划字型交叉识别码

当一个键外字的字根不足四个时，依次键入字根码后，最后补一个识别码，识别码用末笔划的类型编号和字型编号组成。具体地说，识别码后为两位数字，第一位是末笔划类型编号（横 1、竖 2、撇 3、捺 4、折 5），第二位是字型代码（左右型 1、上下型 2、杂合型 3）。把识别码看作一个键的区位码，就会得到交叉识别码（字母码），其码表如下：

末笔划、字型交叉识别码表

	左右 1	上下 2	杂合 3
横 1	11G	12F	13D
竖 2	21H	22J	23K
撇 3	31T	32R	33E
捺 4	41Y	42U	43I
折 5	51N	52B	53V

请看下面的实例：

字	字根	字根码	末笔代号	字型	识别码	编码
苗	艹田	AL	一1	2	12F	ALF
析	木斤	SR	丨2	1	21H	SRH
灭	一火	GO	丶4	3	43I	GOI
赤	二小	FI	丶4	3	43I	FII
迫	白辶	RP	一1	3	13D	RPD

编码中加了识别码后仍不足四码时，击空格键补足。单笔与字根相连的字型为杂合型。加识别码的作用是减少重码，加快选字。在不用识别码时，旮、旯、旭这三个汉字的编码是相同的，也就是重码，加了识别码后就可以分开它们了。

关于末笔划有以下的规定，可以使取码简单、明确。

（1） 末字根为“力、刀、九、七”等时，一律认为末笔划为折。

（2） 象进、逞、远等字，不以“走之”的末笔为末笔（书写时确实是末笔，但是这样的话末笔都一样，减少了识别信息量），约定以去除掉“走之”的末笔为整个字的末笔来构成识别码。进、逞、远的识别码分别为：23；K、13；K、53，V。当以“走之”的末笔为末笔时，它们的识别码都是 43，I，以至无法识别。

（3） 我、成等字的末笔为“丿”。

关于字型有如下约定：

（1） 凡单笔划与字根相连或带点结构都视为杂合型。

（2） 字型区分时，也用“能散不连”的原则。

（3） 内外型字属杂合型，如困、同、匝。但“见”为上下型。

（4） 含两个字根且它们相交者属杂合型。

（5） 下含“走之”的字属杂合型，如：东、串、电、本、无、农、里。

（6） 以下各字为杂合型：司、厅、龙、尼、式、后、反、处、办、皮、习、死、疗、压，但和它们相似的右、左、有、看、者、布、包、友、冬、灰却视为上下型。

3. 字根区位码输入

每个字根都有所在的区位码，在字根总表中已经标明了。例如：王的区位码为 11，白的区位码为 32。一个汉字可用上述的方法按照编码击英文字母键输入汉字，也可以把每个字母换成两位的区位码，击右部数字专用键盘的数字键输入。例如：

字	拆　分	字母码	数 字 区 位 码
地	土也	FBN	12　52　51
量	日一日土	JGJF	22　11　22　12
度	广廿又	YACI	41　15　54　43
重	丿一日土	TGJF	31　11　22　12
国	囗王丶	LGY	24　11　41
码	石马	DC	13　54

区位码不足八位的，击空格键表示编码结束。

九 简 码 输 入

上节所介绍的汉字的字母码，一律码长为四（字根数大于等于四的用四个字母码；字根为三的补一个识别码也为四个字母码；字根数为二的补一个识别码，再补上一个空格键就仍是四键；键面字也一律用四码），为了简化输入，减少码长，设计了简码输入法。简码分一、二、三级，分别只需击一个、两个、三个字母键再击一个空格键来输入简码汉字。由于键盘上的字母键只用到了 25 个，那么，一级简码字可以有 25 个，二级简码字可以有 25×25 个，三级简码字就可以有 25×25×25＝15625 个，然而经过实用性筛选实际上三级汉字只安排了约 4400 多个，简码字总数约为 5000 个。

1　一级简码

一级简码，也常称为高频码。五笔字型中，从 11～55 共 25 个键位代码，根据每键位上的字根形态特征，每键安排一个常用的高频汉字，这类字只要击键一次，再加击一次空格键，即可输入，这些高频字及编码如下：

一 11（G）　地 12（F）　在 13（D）　要 14（S）　工 15（A）
上 21（H）　是 22（J）　中 23（K）　国 24（L）　同 25（M）
和 31（T）　的 32（R）　有 33（E）　人 34（W）　我 35（Q）
主 41（Y）　产 42（U）　不 43（I）　为 44（O）　这 45（P）
民 51（N）　了 52（B）　发 53（V）　以 54（C）　经 55（X）

2　二级简码

二级简码字的简码的前两位相同，即只用前两个字根编码。二级简码字表如下：

GFDSA　HJKLM　TREWQ　YUIOP　NBVCX
11——15　21——25　31——35　41——45　51——55

G	11	五于天末开	下理事画现	玫珠表珍列	玉平不来	与屯妻到互
F	12	二寺城霜载	直进吉协南	才垢圾夫无	坟增示赤过	志地雪支
D	13	三夺大厅左	丰百右历面	帮原胡春克	太磁砂灰达	成顾肆友龙
S	14	本村枯林械	相查可楞机	格析极检构	术样档杰棕	杨李要权楷
A	15	七革基苛式	牙划或功贡	功匠莱共区	苏燕东　芝	世节切芭药
H	21	睛睦　盯虎	止旧占卤贞	睡　肯具餐	眩瞳步眯瞎	卢　眼皮此
J	22	量时晨果虹	早昌蝇曙遇	昨蝗明蛤晚	景暗晃显晕	电最归紧昆
K	23	呈叶顺呆呀	中虽吕另员	呼听吸只史	嘛啼吵　喧	叫啊哪吧哟
L	24	车轩因困	四辊加男轴	力斩胃办罗	罚较　　边	思　轨轻累
M	25	同财央朵曲	由则　崭册	几贩骨内风	凡赠峭　迪	岂邮　凤
T	31	生行知条长	处得各务向	笔物秀答称	入科秒秋管	秘季委么第
R	32	后持拓打找	年提扣押抽	手折扔失换	扩拉朱搂近	所报扫反批
E	33	且肝　采肛	胆肿肋肌	用遥朋脸胸	及胶膛　爱	甩服妥肥脂
W	34	全会估休代	个介保佃仙	用遥朋脸胸	信们偿伙	亿他分公化
Q	35	钱针然钉氏	外旬名甸负	儿铁角欠多	久匀乐炙锭	包凶争色
Y	41	主计庆　度	让刘训为高	放诉衣认义	方说就变这	记离良充率
U	42	闰半关亲并	产间部曾商	产瓣前闪交	六立冰普帝	记离良充率
I	43	汪法尖洒江	让刘训为高	少泊肖兴光	注洋水淡学	沁池当汉涨
O	44	业灶类灯煤	粘烛炽烟灿	烽煌粗粉炮	米料炒炎迷	断籽娄烃
P	45	定守害宁宽	寂审宫军宙	客宾家空宛	社实宵灾之	官字安　它
N	51	怀导居　民	收慢避惭届	必怕　愉懈	心习悄屡忱	忆敢恨怪尼
B	52	卫际承阿陈	耻阳职阵出	降孤阴队隐	防联孙耿辽	也子限取陛
V	53	姨寻姑杂毁	旭如舅	九　奶　婚	妨嫌录灵巡	刀好妇妈姆
C	54	对参戏	台劝观	矣牟能难允	驻　　驼	马邓艰双
X	55	线结顷　红	引旨强细纲	张绵级给约	纺弱纱继综	纪弛绿经比

3　三级简码

三级简码字字数多，输入三级简码字也需击四键（含一个空格键），三个简码字母与全码的前三者相同，但用空格代替了末字根或代替识别码。

三级简码看上去击键次数虽仍为四键，没有减少总的击键次数，但由于省略了前三个字根之后的字根判定或者交叉识别码的判定，所以提高了编码速度。

十　词语输入

在汉字输入方法中，以词语为单位的输入方法通常可达到减少码长，提高效率的目的。在五笔字型输入方法中也设计了词语的输入方法，并给出开放式结构，以利于根据需要自行组织词库。五笔字型词语输入还有一个特点，即词语输入和单字输入统一，不加字或词的输入区分标记也无需用特殊的键换档。这是由于词语的编码也是四码。全部四码空间的大小约 39 万，而一二级汉字单字编码（含简码）共 1·2 万左右，大量编码空间空闲着。词汇码绝大部分都插在了闲置空间中。

然而单字码与词汇码虽有着很不相同的分布规律，但二者却能混在一起而不用换档区分，且绝大多数情况下是不会发生冲突的。单字与词汇编码可以共存共容互不影响，词汇码的输入和单字码的输入可以混合进行。记得住词汇编码的就打词汇以求其快，记不清的仍打单字以求其准。二者之间不需要任何的换档操作。这种设计在实际应用中，给操作者带来了极大的方便。

1 二字词

二字词的词语由所含的两个汉字各取两个字根码组成，即每个字按笔顺有取前两个字根为编码，如：

简化：竹门亻匕 TUWX　　继续：纟米纟十 XOXF

计算：讠十竹目 YFTH　　许多：讠丿夕夕 YTQQ

2 三字词

前两个汉字各取一码，最后一字取前两码。如：

电冰箱：日冫竹木 JUTS　　电视机：日礻木几 JPSM

操作员：扌亻口贝 RWKM　　洗衣机：氵亠木几 IYSM

3 四字词

四字词的词语由每个汉字的第一码组成词语的输入码。如：

知识分子：矢讠八子 TYWB　　日理万机：日王厂木 JGDS

社会主义：礻人丶丶 PWYY　　程序设计：禾广讠讠 TYYY

4 多字词

超过四个字的词，前三个字各取第一个字母码，词语码的第四码是由最末一个汉字的首码组成。换句话说：是由一、二、三和末四个汉字的第一个字母构成的。如：

有志者事竟成：厂士土厂 DFFD

汉字输入技术：氵宀车木 IPLS

十一 重码和容错码的处理

1 重码处理

从输入、打字的要求看，键位要尽量少，码长要尽量短，重码也要尽量少，这自然不是一件容易的事。五笔字型方案中对重码字也用在提示行中编号显示的办法，让用户按最上排的数码键选择所用的汉字。为了提高速度，还做了以下处理：

1. 当屏幕编号显示重码字时，按字的使用频率排列，高频字在第一号位，当高频字排在1号位时，还响铃报警，此时，只要继续输入下面一个字，1号字就会自动跳到屏幕光标处，这样可以减少一次数码选择的操作。

2. 对于国标一级汉字中的重码，把常用的字仍按常规编码，对较不常用的那个把末码改为L，做为一个容错码，使一级汉字中的重码字，大多可以实现无重码输入。

2 容错码

容错码的"容"有两种含义，一是"容易"搞错的容，另一个是"容许"搞错的容。在实际编码中常会出现种种差错，许多差错的产生有其原因，带有普遍的易发性。容错码的设计是一种

"因势利导"的办法，即承认那些容易写错的码产生的合理性，把它们做为一类正常的可以用的码保留。使那些和规则不完全相符的（有错误的）码也可以正常使用，从而简化过程，避免键入错码后找不到所需的字，费时费力地重打。五笔字型软件的第四版中有容错码 500 个，有以下类型：

1. 拆分容错

动：二厶　力　（对）　　一　云　力（容错）

2. 字型容错

右：𠂇口　12（对）　　𠂇　口　13（容错）

3. 软件版本容错

已熟悉老版本打法的用户可把老版本中已改掉的码作为容错码输入。异体容错

按字根：一　冂　丨　辶　可出"迎"。

4. 末笔容错

"化"字末笔可取折（L）也可取为撇（丿）。

5. 笔顺容错

"长"字可按三种笔顺输入均可。

6. 简繁容错

如按繁体字输入"国"，需取字根为：口　戈　　口　一　亦可。

7. 低频重码字后缀

前面重码处理中谈到，把重码字低频字的末码字改为 L，这也可做为一个容错码看待。如：喜、嘉的正常编码是 FKUK，嘉为相对频率低的，就定义编码 FKUL 为嘉字的容错码。

十二　选择式易学输入法

在学习了键盘区位表之后，你可能会问：26 个英文字母键只用了 25 个，还有一个"Z"键为什么闲着不用呢？

原来，我们给"Z"键派了很重要的用场。这就是用它来进行"选择式易学输入"。因为是用"Z"键实现的，有时也叫"Z"功能或"Z"处理。

当由于对键盘字根不太熟悉或者对某一汉字的拆分一时难以确定时，一切是"未知数"的字根都可以用"Z"来代表。在一个汉字的字根输入中，不知道是第几个字根，都可以打"Z"键代替。计算机的软件设计可以帮助检索出那些符合已知字根代码的字，将汉字及其正确代码按两个一组显示在提示行里，根据这些字在提示行里从左到右的位置号，打键盘上的数字键 1、2，即可从提示行中选出需要的字。由于提示行中的每个字后面都显示有它的正确代码，供你学习、掌握。

例如：要打入一个"敬"，而又记不清第二个字根该怎么打，这时可以打"艹 Z 口攵"这样四个键，结果提示行中显示出"敬 AQKN"，这表示符合刚刚打入的字根组合的字只有一个"敬"。只要再按一下数字键"1"，"敬"字就自动显示在正常编辑位置上。同时，可以从提示行中知道，刚才那个未知的字根"勹"在"Q"（即 35）键上。

在打入"宫"字时，如果打了"宀口口"之后对于最后的识别码含糊不清了，这时，也可以用"Z"来代替之。结果，提示行中也只有一个"宫"字，而且提示，最后的识别码为 F（12）。这个例子，也许会给你一个重要的提示：如果对于本方案的"识别码"觉得使用不方便的话，就不妨把全部的"识别码"都打成"Z"，照样可立即在提示行中找到所要找的字。既然如此方便，又何必学习识别码呢？区别在于：取消识别码的话，每个字至少要多打一次键才能选择出来，而每

个字多打一键，是要影响速度的。

未知的字根越多，选择的范围越广。提示行里一批只显示两个字，如果这一批字中没有需要的字，可按空格键就再显示出下一批两个来。如果对某个汉字的四个字根一无所知，就将四个代码都打成“Z”，那么机器就会将 6763 个汉字从头到尾，顺序地分批显示出来。

选择式输入还有一个特色是这样的：如果在四个码中未知的码只有一个的话，允许把这个未知码随便打在第几个位置上都行，都可以很快找到所要的字。举例来说，对于“照”字，假设现在的未知字根是“刀”，那么可以有以下四种键入查找方法：

（1）日 Z 口灬　　（2）日口 Z 灬　　（3）日口灬 Z　　（4）Z 日口灬

然而，以上每一种打法，都只显示“照”和“煦”两个字，当然极易选择。这样处理的好处是：只管顺序打入有把握的字根，而把未知的放到最后打就行了，所以大都采用第（3）种方法。

选择式输入的另一个特点是：所有符合已键入字根的字，基本上按字的使用频率顺序显示出来的。比如，先是高频字，后是二级简码，再是全码字。这样，一般情况下选择不了几下，就可以选到所需的字了！

至此，我们主要讲述了五笔字型输入法的基本规则和使用，对于您来说，更重要的是练习、练习、再练习。学会五笔字型并不难，可是要想达到较高的输入速度、学而好用，还需要下一番功夫。

练 习 一

一 填空

1 五笔字型的字根键盘可分为（　　）个区，每个区分为（　　）个位。

2 字根所在的区号，一般都与字根的第（　　）个笔划的代号一致。

3 在保证字根最佳组合的前题下，字根所在的位号，尽量与它的第（　　）个笔划保持一致。

4 单笔划及复合笔划“一、二、三”，“丿、彡”等，它们的笔划个数与所在键的（　　）号一致。

5 “车、力、几、心、耳”等，它们的首笔代号与它们所在的（　　）号不一致，因而是少数几个例外。

6 一般来说，五笔字型字根的前两笔代号、所在键的（　　）号，以及打键时手指的自然序号，三者是保持一致的。因此，容易做到看到字根，就知道哪个手指，同时也就知道按哪个键。

二 对以下的 26 个常用的偏旁部首做结构分析，并完成后面的填空题。

a 尢　b 寸　c 攵　d 久　e 刀　f 力　g 马　h 尸

i 户　j 衤　k 韦　l 犬　m 歹　n 戈　o 弋　p 瓦

q 攴　r 父　s 手　t 爪　u 欠　v 广　w 疒

1 以上部首中，编号为（　　）的都是五笔字型的基本字根。

2 以上部首中，编号为（　　）的都是含两个字根的非基本字根。

3 在所列非基本字根中，含三个字根的有（　　），含四个字根的有（　　）。

4 在第 2 题中属于散关系的有（　　），属于连关系的有（　　），属于交关系的有（　　）。

5 在第 3 题中属于散关系的有（　　），属于连关系的有（　　），属于交关系的有（　　）。

三 分析题后给出的汉字的字型，填空完成下列各题

在这些汉字中，属于左右型的有（　　）个，属于上下型的有（　　）个，属于杂合型的有（　　）个。

无　卡　严　矢　汉　进　反　左　右　友　京　要　种　年　同

练 习 二

一 填空

1 有两笔划的成字字根的编码公式为（　　）。

2 有三笔或三笔以上笔划的成字字根的编码公式为（　　）。

3 含四个或多个字根的键外字的编码取其（　　）个字根的编码组成。

4 不足四个字根的汉字，其编码由（　　）和（　　）、最后补一个（　　）组成。

5 一级简码字共有（　　）个，二级简码字共有（　　）个，三级简码字最多可为（　　）个，现在实际给出了大约（　　）个。

二　判断下列论述的正误，正确的标 yes ，错误的标 no。

1　输入含两笔的成字字根最少可击键三次。　（　　）

2　输入含两个字根的键外字最少击键四次。　（　　）

3　输入一级简码字，只需要击一键。　（　　）

4　输入三级简码字，必须击四键。　（　　）

三　写出下列各字的五笔字型编码

字	字根	编　码	字	字根	编码
于			午		
牛			年		
矢			朱		
未			末		
万			尤		
夫			元		
平			夹		
书			专		
毛			才		
出			世		
长			垂		
重			曲		
面			州		
发			严		
承			离		
凹			凸		
民			切		

四　操作实习

1　将给出的短文用五笔字型法按字方式反复输入几遍。

2　重复输入短文，能用简码的用简码。

3　重复输入短文，能用词组输入的用词组输入。

短　　文

机械英文打字机出现后，迅速产生了用于印刷的英文排铸机和用于电报通讯的英文电传机，使西方国家开始了文字处理机械化时代——打字机时代。这个时代，汉字的许多缺点暴露得充分而又严重。可以说机械英文打字机给汉字敲过丧钟，数次把汉字推向被审判、被鞭挞的席位。

电脑以其高速度、高智能拯汉字于水火，为汉字信息处理现代化开僻了广阔的美好的前景。使古老的汉字有可能随着信息化进程而发展。

世界范围都在重新认识和评价汉字。

4　自己选择短文不断练习。

中西文浮点 BASIC 使用手册目录

附　表

1 浮点 BASIC 和 FBASIC 功能比较

1.1 浮点 BASIC 的特点

浮点 BASIC 具有微机上配备的 BASIC 语言的基本证句和部分扩展 BASIC 语句。具有基本语句的 BASIC 被称为标准 BASIC。与 FBASIC 相比，浮点 BASIC 的特点在于，计算能力较强，如整整数的位数为 9 位，实型（浮点）数的范围是$\pm 10^{38}$，可以进行整型、实型数的小数、乘方、开方、三角函数、指数函数、自定义函数的运算，其能力完全覆盖中学教学需求。浮点 BASIC 设计时参考了苹果－Ⅱ的浮点 BASIC，使用操作与苹果－Ⅱ BASIC 基本相同，有兴趣的读者可参考《苹果－Ⅱ BASIC 程序设计》（北京师范大学出版社）一书。裕兴还在苹果－Ⅱ BASIC 的基本上增加了 FBASIC 中的音乐指令 PLAY，有关的音乐设计方法，可参阅使用手册《用游戏机作曲》。

浮点 BASIC 有优点也有不足，与 FBASIC 相比，编辑方式为行编辑，而不是全屏幕编辑，没有十六进制数、不能控制卡通、手柄和屏幕定位等等，这些只能不断进行改进。为了方便有 FBASIC 知识的人学习浮点 BASIC，特把两者的区别列表。

1.2 浮点 BASIC 和 FBASIC 比较

两者在规格上的差别可参考下表：

表 1 浮点 BASIC 和 FBASIC 比较

功　　能	浮点 BASIC	FBASIC	备　　注
字符种类	中、英、数字、符号	英、数字、符号	个别符号不同
整数（十进制）	±999999999	±32767	
整数（十六进制）	无	&H0000～&HFFFF	
实数范围	$\pm 10^{38}$	无	
三角函数	有	无	
指数函数	有	无	
乘方开方	有	无	
字符串长度	0～90	0～31	
变量名长度	1～94	255	只识别前两位
行最大容量	98	255	
行号范围	0～9999	0～65535	
一行多个语句	可	可	用冒号隔开
编　辑	行编辑	全屏幕编辑	
画面分层控制	不可 *	可分 4 层	
分辨率	256×240	相同	
色彩控制	不可 *	可	
音　乐	两者相同		
操纵器控制	不可 *	可	
卡通控制	不可 *	可	
打印机	可	不可 *	
注：* 表示暂时未实现			

在运算符号上，两者之间的差别如下表：

表 2　运算符差别比较

运 算 符	浮点 BASIC	FBASIC	备　注
MOD		有	取　余
∧	有		乘　方
XOR		有	异　或

指令之间的差异可用表 3 说明。

表 3　指令之间的差异比较

功　能		浮点 BASIC	FBASIC
系统实用命令	LOAD?		有
	EDIT	有	不需要
	LLIST	有	无
基本指令	LINPUT CLEAR ON SWAP		有
	LPRINT	有	无
屏幕控制指令	除 CLS	无	
MOVE 系列指令		无	有
三角函数	SIN COS ATN TAN	有	
指数对数函数	EXP LOG	有	
开方取整函数	SQR INT	有	
自定义函数	DEF FN(X)	有	
字符函数	HEX $ LEFT $ RIGHT MID $		有
	INKEY $		有
特殊函数	除 FRE、TAB、PEEK 外	无	
卡通控制指令		无	有

2　BASIC 语言基础知识

本章是为想从本书开始学习 BASIC 语言的读者设立的。有关键盘操作和超级学习卡的知识，可参看《F BASIC 操作手册》。有 F BASIC 知识的读者可以跳过这一章。从本章开始，如不特别指明，凡提到 BASIC 这处均指浮点 BASIC。

2.1　BASIC 语言基础知识

BASIC 是十分浒的通用计算机语言，BASIC 语言是一种高级语言，所谓“高级”是指编程的难易而言，用户使用的计算机语言越高级，需要了解计算机的硬件和系统知识就越少，也就越容易学习。BASIC 语言主要有以下几个特点：

1. BASIC 语言简单易学。基本 BASIC 只有 17 个语句，语句和指令近似英语自然语言，而其运算符号和算式同数学上的用法也很类似，故容易理解和记忆。BASIC 程序结构简单，规则也少，易于初学者掌握。

2. BASIC 语言是一种会话式语言，人机对话十分方便。用户可随时中断程序的运行，并从断

点继续执行程序。BASIC 语法检查比较严格，发现错误可通知用户修改。

3. BASIC 语言具有立即执行方式。用这种方式可直接输入指令或语句，按 ENTER 键（回车）后该指令或语句可立即执行。这种方式有利于检查调试程序。

4. BASIC 语言是解释型的，用户编写的程序叫源程序存储在用户程序区中，当键入程序运行指令 RUN 后，用户的程序被逐条的解释，形成目标代码（机器语言）供计算机执行。

顺便说一下，语言并不是越“高级”越好，一些程序员在编程序时，往往希望可直接控制计算机的接口、外设等，像 F BASIC 和 C 语言都提供了有关指令，这类语言往往被称为“中间语言”。所以，在学习计算机语言时一定注意该语言的局限性。

2.2 BASIC 程序的构成

让我们从一个例子说起。

如果 A=1，B=2，求 A+B=？

解：　设 C=A+B

　　　故 C=1+2

　　　即 C=3

把这个例子写成 BASIC 程序如下：

```
>10  LET  A=1
>20  LET  B=2
>30  LET  C=A+B
>40  PRINT “C=”；C
>50  END
>RUN            ；  表示按回车键（Enter 或 RETURN）
C=3             ；屏幕显示结果
```

程序执行的过程是：先让 A 取值 1，再让 B 取值 2，然后让 C 取 A 与 B 之和，再把 C 的值在屏幕上显示（打印）出来，在 50 行程序结束。

上述例子描述了 BASIC 语言的写法（构成）方法。BASIC 语言的构成规则是：

1. 一个 BASIC 程序由若干个程序行组成。上述例子是由五个程序行组成。

2. 每行程序都有行号。上述例子的 10、20、30、40、50 就是行号。BASIC 语言的行号范围是 0～9999。行号可以连续也可不连续。行号之间可以插入新行号，在新行号下写入的新语句，可自动按行号大小顺序插入到相应位置。

3. 每行程序由语句构成。语句又由指令符和指令体构成。上述例子中的“LET、PRINT、END”都是指令符。上述例子中的 A=1、B=2、C=A+B、C 是指令体。如 LET　A=1 即为一个语句，每行程序可由多个语句构成，语句之间用冒号隔开。

4. 指令符作为保留字，不可作为变量名。

5. 每行程序必须“合法”，即符合语法规则。

2.3 常量和变量

1. 常量

在程序执行过程中，常量保持不变。常量又分为三种数据类型：

（一）整数（整型数）

整数的范围是±999999999，在这个数范围之外，只能用实数表示。

（二）实数（实型数）

实数也称浮点数，实数的范围是$\pm 10^{38}$，一个在9位数之内的整数也可用实数表示。如：8888.8，用实数表示可写成8.8888×10^{3}，在BASIC语言中可写成：8.8888E+3。实数表示的基本格式是：

± X.X…X E ± T

- 式中最左边的正号，可以省略。
- X取数字0～9，最大有效位数为9位。
- T取数籽0～9，最大位数为两位。
- E为指数符，代表1×10的方次。

（三）字符串常数

用双引号对“”括起来的字符序列，称为字符串常数。字符串的长度不超过90个字符，汉字不超过45个。如：“裕兴”、“YUXING”、“1A234”、“A12B”、都是合法的字符串常数。当双引号对中内有空格也是字符串常量，称为空串。注意：字符串常量中不要用保留字，也不能用双引号。

2. 变量

在程序运千可取不同值的量，称为变量。

变量在程序运行的不同时刻，可以取不同的值，变量通过变量名表示，变量名的构成遵循如下规定。

（一）变量名长度不许超过90个字符，但只有前两位有效。

（二）变量名的第一个字母必须是英文A～Z中的一个。

（三）保留字不得作为变量名或变量名中的一部分。

变量与常量对应，分为整型变量、实型变量和字符串变量。

整型变量可表示整型数，实型变量可表示实型数，字符串变量可以表示字符串常数，字符变量名的末尾要加$符号。

在任何一个变量参与运算时，必须预先衩赋值，如果预先不赋值，数值变量自动被赋0值，字符串变量自动被赋空格（值）。变量在参与运算时还要注意变量类型，整型变量和实型变量都是数值变量，相互之间可以自行转换，即变量类型相容。而字符串变量属于字符变量，与数值变量不相容。比如A=1，A$ = “1”，两者完全不同，不可以直接进行运算，如要把两者相加，须采用如下方法。

```
>10  A=1: A$ = "1": B=VAL (A$)
>20  PRINT  A+B
>RUN                                 ; 运行程序
2                                    ; 运行结果
```

除上面介绍的变量分类之外，变量还可分为简单变量和数组变量，以上介绍的变量都是简单变量，数组变量又称下标变量，可以是字符型的也可是数值型的。数组变量相当简单变量的有序集合。比如：A（I），当I取1，2，3时，即A（1），A（2），A（3），下面例子可以说明这一点：

```
>10  A (1) =1: A (2) =2: A (3) =3
>20  A1=1: A2=2: A3=3
>30  B=A (1) +A (2) +A (3)
```

```
>40  C+A1+A2+A3
>50  PRINT  B, C
>RUN
6          6
```

可以看出，数组变量和简单变量可以相互替代，功能等效。

数值数组表示为：

A (I)　　　　一维数组（其中 A 为变量名，I 为变量）

A (I, J)　　　二维数组

A (I, J, K)　　三维数组

字符数组变量

A$ (I), A$ (I, J), A$ (I, J, K)

与数值变量不同的是字符数组变量的取值是字符串常量。

尽管数组变量很方便，并可取代简单变量，但是数组变量的最大特点是要占用内存，BASIC 中规定了数组的最大取值个数。

在 32K 用户内存之下，一维数组最大为 A (255)，二维数组最大为 A (80, 80)，三维数组最大为 A (18, 18, 18)。在实际编程时二维、三维数组还不能使用到上述值，如用到上述值，就没有程序空间而无法编程了。

2.4 函数

在 BASIC 程序中，除允许使用常量、变量之外，还允许使用事先规定好的函数，用户只要给出函数名与参数，就可求出相应的函数值。如求 5 的平方概括，可调用函数 SQR (5)。

函数名也是保留字，不可作为变量名。

函数分为数值函数、字符函数和特殊函数。

数值函数有三角函数、指数函数、对数函数、随机函数等。

字符函数有类型转换函数、字符串长度函数等。

特殊函数有 FRE、TAB、PEEK 函数，FRE 用于观察内存量，TAB 用于格式打印，PEEK 用于读内存数据。

2.5 运算符和表达式

运算符分为算术运算符、关系运算符和逻辑运算符。当把数据类型相同的常量、变量和函数用规定的运算符连接起来就构成 BASIC 表达式，分述如下：

1. 算术运算符和表达式

算术运算符有 5 个：

+，－，*（乘），/（除），↑（乘方用∧代替）

算术表达式如：

5*A, 10/5

SIN(3.1416)+15　;SIN(X)是三角函数 sin(x)

2. 关系运算符和表达式

关系运算符是用于比较两个常数和两个变量的符号。

关系运算符有：

关系符	含义
＝	等于
＜ ＞	不等于
＞	左边大于右边
＞＝	左边大于等于右边
＜	左边小于右边
＜＝	左边小于等于右边

关系表达式如：

A＞B，C＜＝D

3. 逻辑运算符和表达式

逻辑运算符有：

逻辑符	含义
NOT	非
AND	与
OR	或

逻辑表达式有：

NOT A， A AND B，A OR B

下表解释了逻辑运算符的含义

表 4 逻辑运算符的含义

A	B	NOTA	A AND B	A OR B
0	0	1	0	0
0	1	1	0	1
1	0	0	0	1
1	1	0	1	1

使用逻辑运算符必须注意逻辑关系是否成立，因为符合 BASIC 语法（规则）的语句，在逻辑关系上并不一定正确。如：

IF A＜0 AND A＞10 THEN 1000

这句的意思是，如 A＜0 与 A＞0 成立，则程序跳转到 1000 行。可以看出既要小于 0 又要在于的数是不存在的。但是从语法上讲，这条语句在 BASIC 语言中是合法的。

4. 字符串运算符与表达式

字符串运算符只有一个，即逻辑加：

逻辑加	含义
＋	将两个字符串连接起来

字符串运算的“加号”，不同数学中的加号，下面的表达式是合理的：

A＄＋B＄，A＄＋“ASD”

2.6 几个具有特殊意义的标点符号

1　—连接号

在 LIST 指令中使用连接号，可把任意一段程序调出来。如：

LIST　100—200　；调出 100—200 行的语句

2　，逗号

逗号用于分割数据，下面的语句是合法的。

例 1：　　INPUT　A，B，C

例 2：　　PRINT　A，B

例 3：　　DATA　10，20，30

3　：冒号

在同一行号下写入一条以上的语句时，两条语句之间用冒号隔开。如：

10　A=B-C：PRINT　A

4　；分号

分号用于分割数据，与逗号分割数据不同，如在打印语句中，用逗号隔开，打印结果之间空 7 格，用分号隔不空格。

5　？号

用问号代替 PRINT

？A，B，C 等效于 PRINT　A，B，C

2.7　框图

框图是编写程序的工具，优点是直观、清晰、易懂，便于检查和交流。框图为了粗框图和细框图，粗框图用于规定程序的功能模块，细框图用于编程。对于一个大型程序，框图必不可少，小程序可以不用框图，但是要养成框图习惯，有利于编写程序的条理性和正确性。分析别人程序往往也要画框图，有利于分析借鉴。框图的规定是约定俗成的，可分述如下：

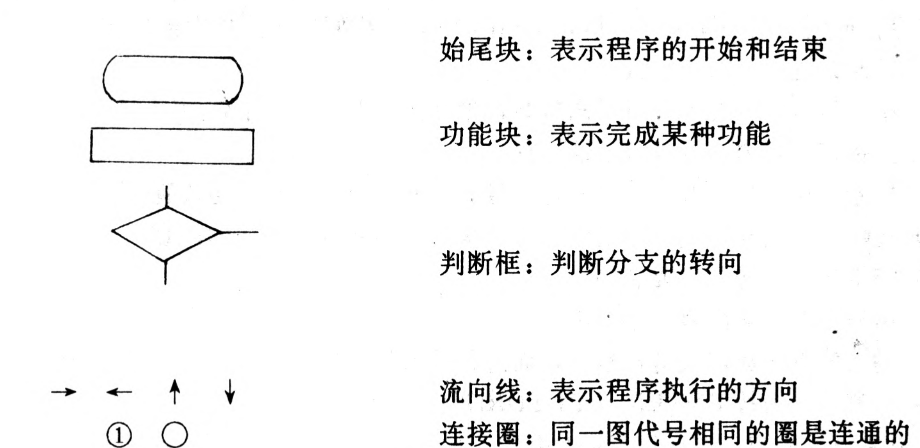


图 2.1　框图的规定

2.8　编写程序的基本方法

编写程序和画画一样，也有好坏高于低手之分。一个优秀的 BASIC 程序，在同等功能的条件下，应该结构清晰，可读性强，程序相对较短。应注意以下三点：

注意目标树的划分，注意变量名的设计，注意子程序的使用。

1. 目标树的划分

编程序的目的是解决特定问题，这个问题就是要解决的目标，究竟怎样解决方法很多，即实现目标的途径不唯一。从编程的角度来看，通常将一个问题分解成一个目标树，如图所示：可以看出，目标树是分层的，同一层代表级别相同和相近的功能，第一层各个节点之下有第二层，第二层对第一层的节点有从属关系。即，第二层节点是第一层相应节点的子功能。这种从属关系可以一直递推。但是，对 BASIC 程序，这种分层层不宜太多，一般以 2～3 层为宜。

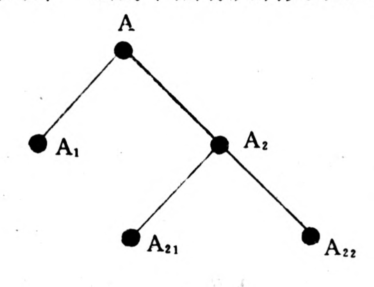


图 2.2　目标树示意

2. 变量的命名

变量的命名应根据目标树进行。比如，第一层用 A，B，第二层用 A1，A2…。按理第三层应用 A11，A12，由于 BASIC 只识别前两痓，故第三层只能采用其他命名法，比如，SA11，AA12。从这里可以看出分层太多会在变量命名方面带来困难。一般，在第三层安排一些公用子程序。

3. 子程序的设计

对不同子目标 A、B，可能在编程序时有很多相同的程序，把这些相同的程序抽出来，写成公共子程序，供不同的程序调用，这样可以缩短程序长度，使程序结构清晰易读。BASIC 语言设计了专门调用子程序的指令：GOSUB，和子程序返回指令：RETURN。

2.9　指令的描述方法

在介绍指令的功能之前，先介绍本书怎样描述指令。对所有的指令都采用功能、格式、缩写、解释、程序例的顺序描述。各部分的作用是：

功　能：解释指令的功能。

格　式：说明指令构成语句的书写格式，格式采用的符号和字母遵循下述规定：

（一）英文大写字母可直接键入

（二）中括弧 [] 内的内容可任意省略，中括弧不要输入到语句中

（三）大括弧 {} 内的内容可任选一项

（四）括号、逗号、冒号、分号、连接号等符号，必须用在指定位置

（五）省略号……表示在一行范围内可多次重复相同或有规律的内容

（六）空格作为符号，不要省略

（七）引号“”中的内容是字符串

（八）键入的程序最左边有符号＞，而运行结果最左边没符号

（九）↓符号代表按 ENTER 或 RETURN 键

其他有关语法内容，在介绍指令时介绍。

解　释：详细说明指令功能或注意事项。

程序例：知觉的程序例子和执行结果。由于程序可能在一行内显示不下，这时应不改行号，连续把一行程序输入完。

缺省值：如果没有给指令指定参数，将自动以缺省值作为参数处理。

注　释：程序注释被写在程序的旁边，用分号隔开

3 BASIC 指令和用法

本章系统地叙述指令的用法供读者查阅。本章采取的写法与 F BASIC 手册相同。

3.1 系统实用命令

NEW

功　能：清除内存区中所有变量和程序。

格　式：NEW

解　释：执行 NEW 指令将清除用户区内存已有的程序（用户自编的程序）。用户在输入新程序前，最好先执行 NEW 指令，可避免新旧程序混杂。

CLS

功　能：清除屏幕并使光标返回左上角。

格　式：CLS

解　释：执行 CLS 后用户程序不被清除，只是清除屏幕。按 HOME 键同执行 CLS 作用相同。

LIST

功　能：把内存中的程序按行号大小，在屏幕上列出。即列程序清单。

格　式：LIST　[m][－ [n]]

解　释：LIST 指令的作用是列程序清单，当程序被列出后，可较容易发现错误从而有利于修改。利用 LIST 指令，可任意列出所想看的程序。说明如下：

```
LISTm          ；列出第 m 行的程序
LISTm－        ；列出从 m 行到结束行的程序
LISTm－n       ；列出从 m 行开始到 n 行结束的程序
LIST－n        ；列出 n 行之前的程序。
```

程序例：参见 F BASIC 使用手册第四章 LIST 指令的例子。

LLIST

功　能：把内存中的程序在打印机上打印出来。即列出程序清单。

格　式：LLIST　[m][－n]]

解　释：LLIST 的操作与 LIST 相同，不同的是前者将程序清单列到打印机上，后者将程序列到屏幕上。

RUN

功　能：运行内存中的程序。

格　式：RUN [m]

解　释：（一）　如仅键入 RUN，从头开始执行程序。

（二）　如键入 RUNm，从 m 行开始执行程序。

程序例：

```
>10  REM * RUN *              ；10－40 行是输入的程序
>20  PRINT “FAMILY”           ；分号可使打印结果联在一起
>30  PRINT “BASIC”
>40  END
```

```
>RUN                               ；从头开始运行程序
FAMILY BASIC                        ；运行结果
>RUN  30                           ；从30开始运行程序
BASIC                              ；运行结果
```

STOP

功　能：中断程序运行。

格　式：STOP

解　释：（一）在调试程序时，STOP 指令很有用。在程序中插入 STOP 指令，程序运行到有 STOP 指令的行时自动停止。并显示：

“BREAK　IN　××”　　　　　　；××表示行号

这时可以看程序运行结果和预想是否一致，若没问题，在用 CONT 指令继续运行程序。

（二）键盘上有 Pause 键，产生的中断效果同 STOP 指令。

程序例：参见 CONT 指令。

CONT

功　能：让被中断的程序继续执行。

格　式：CONT

解　释：（一）当程序被 STOP 中断时，可用 CONT 继续执行。

（二）当程序出错给出错误信息时，用 CONT 从出错行的下一行继续执行程序。

程序例：目的是了解 STOP 和 CONT 的使用，其他指令不理解没关系。

例 1：
```
>10  REM * STOP * CONT *
>20  FOR  I=1  TO  10
>30  PRINT  I
>40  STOP                         ；程序运行到这一行停止
>50  NEXT  I
>RUN
1                                 ；显示结果
BREAK  IN  40                     ；程序打断在40行
>CONT                             ；键入CONT
2                                 ；运行结果
BREAK  IN  40                     ；程序打断在40行
```

例 2：
```
>10  PRINT “A”
>20  PTIN “A1”
>30  PRINT “A2”
>RUN
A                                 ；运行结果
SYNTAX  ERROR  IN  20             ；语法错误在20行
>CONT                             ；继续程序执行
A2
```

END

功　能：使程序结束运行。

解　释：END 常用在判断语句中让程序结束。如程序在最后一行结束可以不用。

程序例：

```
>10  REM * END *
>20  INPUT  A
>30  PRINT  A
>40  IF  A=10  THEN  END      ；程序结束在 40 行
>50  GOTO  20
>RUN
? 1                            ；输入 1
1                              ；打印结果
? 10                           ；再次显示问号，要求输入
10                             ；程序结束，不再显示问号
```

CLEAR

功　能：清除内存所有变量的值。

格　式：CLEAR

解　释：数字变量清除后值为 0，字符变量变为空。

程序例：

```
>10  REM * CLEAR *
>20  Y=178
>30  A$ = "ASC"
>40  PRINT  Y, A$
>PRINT  Y, A$
178        ASC                 ；运行结果
>CLEAR
0                              ；变量值被清除
```

EDIT

功　能：按行编辑修改程序。

格　式：EDIT　行号

解　释：本指令只可编辑行号所指定的行。用方向键、删除键和空格键进行修秋，完毕后键入 ENTER 键。

SAVE

功　能：把内存的内容存到磁带上。

格　式：SAVE"文件名"

解　释：内存中的程序，不能长期保存，需要长期保存时可用 SAVE 指令存到磁带上，每次存入的程序形成一个文件。

程序例：

```
>SAVE "TEST"                   ；存入文件
                               ；正在存入
```

>■　　　　　　　　　　；存入结束显示光标

LOAD

功　能：将磁带上的文件调回（输入）到内存。

格　式：LOAD“文件名”

解　释：（一）文件名是由 SAVE 指令指定的文件名。

（二）执行 LOAD 指令将清除内存原有的程序和变量。

程序例：

>LOAD

；取出磁带上的第一个文件

>■　　　　　　　　　　；调出结束

AUTO

功　能：行号自动增加。

格　式：AUTO　[m]，[n]

解　释：（一）执行 AUTO 指令，程序行号可自动增加不用键入。

M：超始行号　　　　　N：行号增量

（二）AUTO　M　　　　；从第 M 行开始，行号增一为 10

（三）AUTO，N：　　　　；从第 M 行开始，行号增量为 N

（四）AUTO，N　　　　；从当前行开始，行号增量为 N

（五）AUTO　　　　　　；起始行号为 10，行号增量为 10

（六）在当前行号下不输入内容，按 ENTER 键可从 AUTO 状态退出。

SYS

功　能：用于断电保持

格　式：SYS

解　释：（一）在每次按游戏机复位键和关机之前，打入 SYS 可保持用户程序不丢失。

（二）在 F BASIC 中执行此指令，返回上级菜单。在 BASIC 中执行此指令显示“断电保持”，用 Pause 键可退回系统。

（三）要此指令有效，卡内必须装两节 7 号电池。

EXEC

功　能：执行内存中的程序。

格　式：EXEC\文件名

解　释：本指令对中西文都有效。不同类型的文件，用扩展名区别。特定的文件只能在特特定的语言环境中使用，如语言环境不对，程序不执行。扩展名含义是：

EXE　　；系统可立即执行文件，可在 BASIC 状态下执行

BS　　；F BASIC BS V2.1 源程序文件

BSE　　；BS 可立即执行文件

V3　　；F BASIC PLUS 和 V3 版源程序文件

V3E　　；V3 可立即执行文件

BAS　　；BASIC 源程序文件

BAE　　；BASIC 源程序文件，可立即执行

ASM　　；汇编语言文件

COM　　　；系统文件

LIB　　　；库文件

LOG　　　；LOGO 语言文件

C　　　　；C 语言文件

文件名可用 4 位字符，扩展名为 3 位字符，中间用点“.”隔开。格式是：

文件名．扩展名

扩展名最后一个字符是 E，当程序调入内存后可立即执行。原先存入的程序被清除。

本指令只在 BASIC 状态（系统状态）下执行，如要运行其它语言文件，将自动进入该语言，再把文件调入内存。

当语言环境不对或版本不对时，自动给出错误信息。

程序例：

```
>EXEC\DI                      ；将 DI.BAS 文件调入内存
>■                            ；调入结束
```

DIR

功　能：将文件名列表显示。

格　式：DIR

解　释：这个指令作用同 PC 机 DOS 中的 DIR 指令。

CALL

功　能：在 BASIC 中调用机器码（指令）子程序。

格　式：CALL　地址

解　释：（一）游戏机和中华学习机的机器码基本相同，都是 6502 指令系统。

（二）地址必须是十进制的。

（三）可调用系统中现成的子程序，但必须知道子程序的入口。

（四）自编的子程序必须用 RTS 指令返回。

程序例：

```
>10  REM * CALL *
>20  PRINT "A"
>30  CALL  41501              ；发出“嘟”的一声
>40  GOTO  10                 ；程序只能用 STOP 中断
```

3.2　基本指令

基本指令是编程常用的指令，系统实用指令也可用于编程，与系统实用指令不同的是基本指令如不给参数一般不执行。指令和函数也有区别，指令在行号之后，函数在指令之后，一般函数可以被赋值，而指令不可被赋值。

LET

功　能：给变量赋值。

格　式：LET　变量＝［常数、字母、字符串和表达式］

解　释：（一）赋值是把等号右边的值赋给左边，左边必须是变量。

（二）等号右边如果是变量，必须预先被赋值，未赋值的数值变量自动赋“0”值，未被赋值的字符变量自动赋“空格”值。

（三）等号两边变量类型必须相同，左侧是数值变量右侧必须是数值变量，左侧是字符变量右侧必须是字符变量。

（四）LET 可以省略。

程序例：

```
>10  REM * LET *
>20  A=10                  ；给变量 A 赋值
>30  B=A                   ；给变量 B 赋值
>40  A $ "CHINA  是中国"；给变量 A $ 赋值
>50  PRINT  A，B，A $      ；打印结果
>RUN                       ；运行程序
10  10  CHINA  是中国      ；运行结果
```

PRINT

功　能：将运行结果在屏幕上打印出来。

格　式：PRINT　内容［{;,} 内容］

内容＝常数、变量、表达式、"字符串"

简　写：?

解　释：（一）用 PRINT 可在屏幕上打印出程序运行结果，也可打印出字符串、变量、常数等，在变量、字符串或常数之间插入分号，可以铱次打印，如插入逗号，可按格式打印。

（二）屏幕用于显示打印结果时分为三个区：

一区：1～7 列　　二区：8～15 列　　三区：16～30 列

程序例：

```
>10  REM * PRINT *
>20  PRINT  A；B          ；变量之间用分号
>30  PRINT  A，B          ；变量之间用逗号
>RUN
00                        ；变量之间插入分号的结果
0       0                 ；变量之间插入逗号的结果
```

LPRINT

功　能：将程序运行的结果在打印机上打印出来。

格　式：LPRINT　内容［{;,} 内容］

内容＝常数、变量、表达式、"字符串'

解　释：本指令与 PRINT 指令功能相同，只是将打印结果送到打印机。

程序例：

```
>10  FOR  I=48  TO  119
>20  FOR  J=33  TO  126
>30  LPRINT  CHR $ (27)；CHR $ (I)；CHR $ (J)
>40  NEXT  J
>50  NEXT  I               ；将汉字字库打印出来
```

INPUT

功　能：在程序运行期间从键盘接收数据。

格　式：INPUT ［“字符串”,］［变量，…］

解　释：（一）执行 INPUT 指令，机器等待从键盘接收数据，输入数据后才能执行下面的指令。

（二）当 INPUT 后跟字符串，只显示光标不显示问号，当 INPUT 之后不跟字符串，显示问号。

（三）给多个变量送数，数据可用逗号隔开，也可一个一个输入。

（四）输入错误显示 REENTER，要求重新输入。

程序例：

```
>10  REM * INPUT *
>20  INPUT  A$，A，B      要求输入数据
>30  PRNIT  A$，A，B      打印数据
>40  INPUT "A"，A        指令后跟字符串
>RUN
? A，2，3                 输入数据
A      2      3
A■                        显示"A"，要求输入数据
```

GOTO

功　能：让程序跳转到指定的行号。

格　式：GOTO　n

n：表示行号

解　释：本指令用于改变程序的转向，n 指出了程序的跳转行号。

程序例：

```
>10  REM * GOTO *
>20  INPUT  A
>30  PRINT  A
>40  GOTO  20            ；程序跳回 20 行要求再次输入
```

GOSUB

功　能：转入指定行号子程序继续执行。

格　式：GOSUB　n

n：表示行号

解　释：当一部分程序被频繁调用，若反复写多次将增加程序长度，这时可把频繁使用的一段程序提出来作为公用程序，每次执行这段程序时可用 GOSUB 调用，这种公用程序是子程序。

程序例：参看 RETURN

RETURN

功　能：从子程序返回。

格　式：RETURN

解　释：RETURN 必须与 GOSUB 联合使用，RETURN 只能返回到 GOSUB 语句所在行的下一行。

程序例：

```
>10  REM * GOSUB * RETURN *
```

```
>20  A=A+1
>30  GOSUB  100           ；调用 100 句开始的子程序
>40  B=B+1                ；子程序返回到 40 行
>50  GOSUB  100           ；再次调用子程序
>60  END                  ；子程序回到 60 行
>100 PRINT  A，B
>110 RETURN
```

IF～THEN

功　能：根据条件式的值确定程序的流向。

格　式：IF　条例式　THEN｛行号，语句｝

解　释：（一）条例式是表达式，表达式可以是算术、字符或逻辑式。

（二）行号是所要跳转的行号。

程序例：

```
>10  REM＊IF～THEN＊
>20  INPUT  A
>30  IF  A=999  THEN  PRINT "END"：END
>40  PRINT  A，            ；如果 A 不等于 999，THEN 后的
>50  GOTO  20              ；语句均不执行。
```

FOR～NEXT

功　能：反复执行 FOR 与 NEXT 之间的程序。

格　式：FOR　变量=初始值　TO　结束值　STEP 步长

……

……

NEXT　变量

解　释：(一) 变量为数值变量，称为循环变量。初始值、结束值和步长均为整数。步长可正可负。

（二）省略 STEP 步长，步长为 1，即步长的缺省值为 1。

（三）FOR 表示循环开始，NEXT 表示循环结束，两者需配对出现。如：

```
FOR  I=1  TO  10  STEP  2
PRINT  1
NEXT  1
```

当程序执行到 FOR 语句时，先把循环变量 I 置为 1，到 NEXT　I 语句返回 FOR。这时循环变量按 SIEP 规定的步长增加，STEP2 表示步长为 2，I 增加 2。每循环一次 I 增加 2，当 I 大于 10 后循环结束，执行 NEXT 下一条语句。

（四）多后果循环先内后外，NEXT 后要加变量。

程序例：

```
>10  REM＊FOR～NEXY＊
>20  FOR  I=1  TO  20      ；三重循环
>30  FOR  J=1  TO  I
>40  FOR  K=1  TO  J
>50  PRINT "X"             ；用 X 建立图形
```

```
>60  NEXT  K
>70  PRINT                    ; PRINT 起换行作用
>80  NEXT  J
>90  PRINT
>100 NEXT  I
```

READ

功　能：把写在 DATA 指令的数据读出来。

格　式：READ　变量［，变量，变量，……］

解　释：READ 指令和 DATA 指令配对出现。

程序例：参见 DATA

DATA

功　能：定义读指令所需的数据。

格　式：DATA　常数［，常数，常数，……］

解　释：（一）当常数取汉字时，要在汉字的外边加引号。字符和数字不用加引号。

（二）DATA 语句可放在程序的任意地方。

程序例：

```
>10  REM * READ~DATA *
>20  FOR  I=1  TO  5
>30  READ  A$
>40  PRINT  A$
>50  NEXT  I
>60  DATA "裕兴", "第一", 888, ",", OK
```

RESTORE

功　能：把数据 DATA 中的指针指向第一个数据，使 DATA 数据可以重新使用。

格　式：RESTORE［行号］

解　释：当数据要重复使用，可用本指令改变指针。

程序例：

```
>10  REM * RESTORE
>20  RESTORE  110          ; 指针指向 110 行
>30  READ  A
>40  PRINT  A
>50  RESTORE  110          ; 指针指向 110 行
>60  READ  B
>70  PRINT  B
>100 DATA20
>110 DATA30
```

REM

功　能：注释，供阅读程序清单时参考。

格　式：REM［注释内容］

解　释：（一）注释长度不得超过 94 个字符。

（二）REM 之后的内容不执行。

程序例：

```
>10  REM * REM *
>20  REM 裕兴
>30  REMBASIC
>RUN                        ；运行后什么都没显示
```

DIM

功　能：定义数组。

格　式：DIM 变量名（m1 [，m2 [，m3]]）[，变量名（n1 [，n2 [，n3]]）………]

解　释：（一）数组可以是一维的，可是二维的，也可是三维的。二维以上的数组也叫矩阵。

（二）数组可以数值型的也可是字符型的。

（三）一维数组最大为 M（255），二维数组最大为 M（80，80），三维数组最大为 M（18，18，18）。数组小于 10 可不定义。

程序例：

```
>10  REM * DIM *
>20  DIM  A (20), B (20, 20)
>30  FOR  I=1  TO  19
>40  FOR  J=1  TO  19
>50  PRINT  B (J, 1)
>60  PRINT  A (I)
>70  NEXT  J
>80  NEXT  I
```

POKE

功　能：把数据放入内存单元。

格　式：POKE　内存单元地址，数据

解　释：（一）内存单元的地址从 24576－32767，为 8K。只可在此范围内使用 POKE 指令。

（二）利用本指令可直接修改内存数据。

程序例：参见 PEEK 函数

PALY

功　能：演奏音乐。

格　式：PALY“音乐字符”

解　释：有关 PALY 的使用参见《用游戏机作曲》。

3.3　数值函数

数值函数有三角函数、指数函数、随机函数等，分述如下：

ABS

功　能：取一个数的绝对值。

格　式：ABS(X)

解　释：ABS(X)是绝对值函数,X 可取小数。

程序例：

```
>10  PRINT  ABS(-41)
>RUN
41                ;运行结果
```

SGN

功　能:测试变量的符号。

格　式:SGN(X)

解　释:利用符号函数,可以测量变量的正、负和零。

X>0　SGN(X)=1

X=0　SGN(X)=0

X<0　SGN(X)=-1

程序例:

```
>10  X=41
>20  GOSUB100
>30  X=-41
>40  GOSUB100
>50  X=0 : END
>100 PRINT  SGN(X)
>110 RETURN
>RUN
1     -1     0
```

RND

功　能:让计算机随机(随意)产生数据。

格　式:RND(X)

解　释:RND(X)是随机函数,可在 X 的范围内随机发生 9 位数的数据。

程序例:

```
>10  REM * RND *
>20  X=10
>30  PRINT  RND(X)
>40  GOTO  30
```

INT

功　能:数据取整。

格　式:INT(X)

解　释:当小数位数较多,可用取整函数进行取整。

程序例:

```
>10  REM * INT *
>20  S=4678.234
>30  PRINT  INT(S+0.5)          ;四舍五入
>40  PRINT  INT(S*100+0.5)/100;精确到小数点后两位
>RUN
4678
```

4678.23

SQR(X)

功　能:取一个数的平方根。

格　式:SQR(X)

解　释:取一个数的平方根,X 必须大于零。

程序例:

```
>10  REM * SQR(X) *
>20  PRINT  SQR(2)              ;计算 2 的开方
>RUN
1.41421356                      ;运行结果
```

EXP(X)

功　能:求 e 的 X 次方。

格　式:EXP(X)

解　释:EXP(X)是指数函数,可求 e 的 X 次方。e=2.71828183

程序例:

```
>10  REM * EXP(X) *
>20  PRINT  EXP(10)          ;求 e 的 10 次方
>RUN
22026.4658
```

LOG(X)

功　能:求以自然对数 e 为底的对数 X

格　式:LOG(X)

解　释:LOG(X)是求以自然对数为底的对数,如求以 10 为底的对数,需用换底公式。

程序例:

```
>10  REM * LOG(X) *
>20  PRINT  LOG(10)
>RUN
2.30258509
```

SIN(X)[COS(X)、TAN(X)、ATN(X)]

功　能:求三角函数 sin(x)的值。

格　式:SIN(X)

解　释:三角函数的参数应取弧度值,角度换算成弧度的公式:

弧度=角度 * 3.14159/180

程序例:

```
>10  REM * SIN(X) *
>20  PRINT  SIN(10 * 3.14/180)
>RUN
0.1736
```

DEF　FN

功　能：用户自己定义函数。
格　式：DEF　FN　变量(参数)＝数学表达式
解　释：(一)经过定义的函数才能调用。
　　(二)自定义函数的值为实型量。
　　(三)自定义函数的参数，只许有一个。
　　(四)用 DEF　FN 定义一个函数，如果右部的表达式中有错误，这种错误在首次调用时才能发现，所提出的出错行号为调用该函数的程序行号。

程序例：

```
>10  REM * DEF  FN *
>20  DEF  FN  A(X)=100
* X+5
>30  PRINT  A(1)
>RUN
105                               ;运行结果
```

3.4　字符函数

字符函数用于字符处理，可以进行数值和字符的转换。

CHR $

功　能：把 255 个机内代码转换成所代表的图形。
格　式：CHR $ (X)
解　释：在设计程序时，CHR $ 函数有时可以产生特殊效果。如给输出的字符串带双引号，可利用 CHR $ 函数来实现。

程序例：

```
>10  REM * CHR $ *
>20  PRINT  CHR $ (34);"裕兴";CHR $ (34)
>RUN
>"裕兴"
```

STR $

功　能：把数值数据转成字符型数据。
格　式：STR $ (X)
解　释：有时需把数值型数据转换成字符型数据。

程序例：

```
>10  REM * STR *
>20  A=41:A=41
>30  A $ =STR $ (A)
>40  B $ =STR $ (B)
>50  PRINT  A $ +B $ ;"……";A+B
>RUN
4141……82
```

VAL

功　能:把字符型数字转换成数值型数据。

格　式:VAL(字符变量)

解　释:VAL 的含义与 FBASIC 不同,仅可把用字符串表示的数字转换成数值型数据。

程序例:

```
>10  REM * VAL *
>20  INPUT  A $
>30  A=VAL(A $ )
>40  PRINT  A $ ,A
>50  GOTO  20
```

LEN

功　能:计算字符串的长度。

格　式:LEN(字符表达式)

解　释:在计算字符串长度时,空格和非印刷符也计算在内。字符串长度不超过 94。

程序例:

```
>10  REM * LEN *
>20  A $ =A $ +"A"
>30  PRINT,LEN(A $ ),A
>40  FRE                          ;显示内存剩余容量
>50  A=A+1
>60  GOTO  20
```

3.5　特殊函数

PEEK

功　能:从内存中读数数据。

格　式:PEEK(数值变量)

解　释:(一)PEEK 仅从内存中读数据而不改变内存的值。

　　　　(二)PEEK 和 POKE 的作用相反。

程序例:

```
>10  REM * POKE * PFFK *
>20  D=0
>30  FOR  I=30000  TO  31000
>40  POKE  I,D
>50  D=D+1
>60  RD=PEEK(I)
>70  PRINT  RD,D
>80  NEXT  I
```

TAB(X)

功　能:控制光标在一行的位置。

格　式:TAB(X)

解　释:利用 TAB 可以控制定位打印,TAB 取值为 1－25。

程序例：

```
>10  REM * TAB *
>20  FOR  I=1  TO  25
>30  PRINT  TAB(I);8;
>40  NEXT  I
```

FRE

功　能:确定内存剩余容量。

格　式:FRE

解　释:FRE 实际上是指令,不可把 FRE 赋值给一个变量。

程序例：

```
>10  REM * FRE *
>20  FRE
>RUN
32254                        ;内存有 32254 个字节空余
```

附表 1　BASIC　机内字符代码表

十进制	键　名	十进制	键　名	十进制	键　名	十进制	键　名
0		32	空　格	64	@	96	`
1		33	!	65	A	97	a
2		34	″	66	B	98	b
3	Ctrl+C	35	#	67	C	99	c
4		36	$	68	D	100	d
5		37	%	69	E	101	e
6		38	&	70	F	102	f
7	BELL	39	′	71	G	103	g
8	TAB	40	(	72	H	104	h
9		41	)	73	I	105	i
10		42	*	74	J	106	j
11		43	+	75	K	107	k
12		44	,	76	L	108	l
13	CR	45	—	77	M	109	m
14		46	,	78	N	110	n
15		47	/	79	O	111	o
16	Ctrl+P	48	0	80	P	112	p
17		49	1	81	Q	113	q
18		50	2	82	R	114	r
19		51	3	83	S	115	s
20		52	4	84	T	116	t
21		53	5	85	U	117	u
22		54	6	86	V	118	v
23		55	7	87	W	119	w
24		56	8	88	X	120	x
25		57	9	89	Y	121	y
26		58	:	90	Z	122	z
27	ESC	59	;	91	[	123	{
28	→	60	<	92	\	124	V
29	←	61	=	93	]	125	}
30	↑	62	>	94	Λ	126	×
31	↓	63	?	95	—	127	÷

附表 2　BASIC　出错信息一览表

错误代号	英　文	出错信息释义
	BREAK	打断程序运行
	SYNTAX　ERROR	语法错误
	OUT　OF　MEMORY	内存出界
	CAN' T　CONTINUE	不能继续
	LINE　BUFFER　OVERFLOW	行缓存区溢出
FC	ILLEGAL　FUNCTION　CALL	非法函数调用
OV	OVERFLOW	上溢出
UN	UNDERFLOW	下溢出
IQ	ILLEGAL　QUANTITY	非法数值
EC	EXPRESSION　TOO　COMPLEX	表达式太复杂
UF	UNDEFINED　FUNCTION	调用未定义函数
RD	REDIMENSIONED　ARRAY	数组定义重复
/O	DIVIDED　BY　ZERO	用 0 做除数
TM	TYPE　MISMATCH	数据类型不匹配
LS	LINE　TOO　LONG	一行字符太长
ST	STATEMENT　TOO　LONG	语句太长
NF	NEXT　WITHOUT　FOR	有 NEXT 无 FOR
FN	FOR　WITHOUT　NEXT	有 FOR 无 NEXT
RG	RETURN　WITHOUT　GOSUB	有 RETURN 无 GOSUB
OD	OUT　OF　DATA	数据不够
BL	BAD　LINE　NUMBER	行号错误
BS	BAD　SUBSCRIPT	数组下标错误
IS	INCORRECT　STATEMENT	语句错误
ID	ILLEGAL　DIRECT	不能用于立即执行方式
TP	TAPE　I/O　ERROR	磁带录放错误
	PRT　ERROR	打印机连接错误

国标一级汉字表

0 1 2 3 4 5 6 7 8 9
160 啊阿埃挨哎唉哀皑癌
161 蔼矮艾碍爱隘鞍氨安俺
162 按暗岸胺案肮昂盎凹敖
163 熬翱袄傲奥懊澳芭捌扒
164 叭吧笆八疤巴拔跋靶把
165 耙坝霸罢爸白柏百摆佰
166 败拜稗斑班搬扳般颁板
167 版扮拌伴瓣半办绊邦帮
168 梆榜膀绑棒磅蚌镑傍谤
169 苞胞包褒剥

0 1 2 3 4 5 6 7 8 9
170 薄雹保堡饱宝抱报暴
171 豹鲍爆杯碑悲卑北辈背
172 贝钡倍狈备惫焙被奔苯
173 本笨崩绷甭泵蹦迸逼鼻
174 比鄙笔彼碧蓖蔽毕毙毖
175 币庇痹闭敝弊必辟壁臂
176 避陛鞭边编贬扁便变卞
177 辨辩辫遍标彪膘表鳖憋
178 别瘪彬斌濒滨宾摈兵冰
179 柄丙秉饼炳

0 1 2 3 4 5 6 7 8 9
180 病并玻菠播拨钵波博
181 勃搏铂箔伯帛舶脖膊渤
182 泊驳捕卜哺补埠不布步
183 簿部怖擦猜裁材才财睬
184 踩采彩菜蔡餐参蚕残惭
185 惨灿苍舱仓沧藏操糙槽
186 曹草厕策侧册测层蹭插
187 叉茬茶查碴搽察岔差诧
188 拆柴豺搀掺蝉馋谗缠铲
189 产阐颤昌猖

0 1 2 3 4 5 6 7 8 9
190 场尝常长偿肠厂敞畅
191 唱倡超抄钞朝嘲潮巢吵
192 炒车扯撤掣彻澈郴臣辰
193 尘晨忱沉陈趁衬撑称城
194 橙成呈乘程惩澄诚承逞
195 骋秤吃痴持匙池迟弛驰
196 耻齿侈尺赤翅斥炽充冲
197 虫崇宠抽酬畴踌稠愁筹
198 仇绸瞅丑臭初出橱厨躇
199 锄雏滁除楚

0 1 2 3 4 5 6 7 8 9
200 础储矗搐触处揣川穿
201 椽传船喘串疮窗幢床闯
202 创吹炊捶锤垂春椿醇唇
203 淳纯蠢戳绰疵茨磁雌辞
204 慈瓷词此刺赐次聪葱囱
205 匆从丛凑粗醋簇促蹿篡
206 窜摧崔催脆瘁粹淬翠村
207 存寸磋撮搓措挫错搭达
208 答瘩打大呆歹傣戴带殆
209 代贷袋待逮

0 1 2 3 4 5 6 7 8 9
210 怠耽担丹单郸掸胆旦
211 氮但惮淡诞弹蛋当挡党
212 荡档刀捣蹈倒岛祷导到
213 稻悼道盗德得的蹬灯登
214 等瞪凳邓堤低滴迪敌笛
215 狄涤翟嫡抵底地蒂第帝
216 弟递缔颠掂滇碘点典靛
217 垫电佃甸店惦奠淀殿碉
218 叼雕凋刁掉吊钓调跌爹
219 碟蝶迭谍叠

0 1 2 3 4 5 6 7 8 9
220 丁盯叮钉顶鼎锭定订
221 丢东冬董懂动栋侗恫冻
222 洞兜抖斗陡豆逗痘都督
223 毒犊独读堵睹赌杜镀肚
224 度渡妒端短锻段断缎堆
225 兑队对墩吨蹲敦顿囤钝
226 盾遁掇哆多夺垛躲朵跺
227 舵剁惰堕蛾峨鹅俄额讹
228 娥恶厄扼遏鄂饿恩而儿
229 耳尔饵洱二

0 1 2 3 4 5 6 7 8 9
230 贰发罚筏伐乏阀法珐
231 藩帆番翻樊矾钒繁凡烦
232 反返范贩犯饭泛坊芳方
233 肪房防妨仿访纺放菲非
234 啡飞肥匪诽吠肺废沸费
235 芬酚吩氛分纷坟焚汾粉
236 奋份忿愤粪丰封枫蜂峰
237 锋风疯烽逢冯缝讽奉凤
238 佛否夫敷肤孵扶拂辐幅
239 氟符伏俘服

0 1 2 3 4 5 6 7 8 9
240 浮涪福袱弗甫抚辅俯
241 釜斧脯腑府腐赴副覆赋
242 复傅付阜父腹负富讣附
243 妇缚咐噶嘎该改概钙盖
244 溉干甘杆柑竿肝赶感秆
245 敢赣冈刚钢缸肛纲岗港
246 杠篙皋高膏羔糕搞镐稿
247 告哥歌搁戈鸽胳疙割革
248 葛格蛤阁隔铬个各给根
249 跟耕更庚羹

0 1 2 3 4 5 6 7 8 9
250 埂耿梗工攻功恭龚供
251 躬公宫弓巩汞拱贡共钩
252 勾沟苟狗垢构购够辜菇
253 咕箍估沽孤姑鼓古蛊骨
254 谷股故顾固雇刮瓜剐寡
255 挂褂乖拐怪棺关官冠观
256 管馆罐惯灌贯光广逛瑰
257 规圭硅归龟闺轨鬼诡癸
258 桂柜跪贵刽辊滚棍锅郭
259 国果裹过哈

0 1 2 3 4 5 6 7 8 9
260 骸孩海氦亥害骇酣憨
261 邯韩含涵寒函喊罕翰撼
262 捍旱憾悍焊汗汉夯杭航
263 壕嚎豪毫郝好耗号浩呵
264 喝荷菏核禾和何合盒貉
265 阂河涸赫褐鹤贺嘿黑痕
266 很狠恨哼亨横衡恒轰哄
267 烘虹鸿洪宏弘红喉侯猴
268 吼厚候后呼乎忽瑚壶葫
269 胡蝴狐糊湖

0 1 2 3 4 5 6 7 8 9
270 弧虎唬护互沪户花哗
271 华猾滑画划化话槐徊怀
272 淮坏欢环桓还缓换患唤
273 痪豢焕涣宦幻荒慌黄磺
274 蝗簧皇凰惶煌晃幌恍谎
275 灰挥辉徽恢蛔回毁悔慧
276 卉惠晦贿秽会烩汇讳诲
277 绘荤昏婚魂浑混豁活伙
278 火获或惑霍货祸击圾基
279 机畸稽积箕

0 1 2 3 4 5 6 7 8 9
280 肌饥迹激讥鸡姬绩缉
281 吉极棘辑籍集及急疾汲
282 即嫉级挤几脊己蓟技冀
283 季伎祭剂悸济寄寂计记
284 既忌际妓继纪嘉枷夹佳
285 家加荚颊贾甲钾假稼价
286 架驾嫁歼监坚尖笺间煎
287 兼肩艰奸缄茧检柬碱硷
288 拣捡简俭剪减荐槛鉴践
289 贱见键箭件

0 1 2 3 4 5 6 7 8 9
290 健舰剑饯渐溅涧建僵
291 姜将浆江疆蒋桨奖讲匠
292 酱降蕉椒礁焦胶交郊浇
293 骄娇嚼搅铰矫侥脚狡角
294 饺缴绞剿教酵轿较叫窖
295 揭接皆秸街阶截劫节桔
296 杰捷睫竭洁结解姐戒藉
297 芥界借介疥诫届巾筋斤
298 金今津襟紧锦仅谨进靳
299 晋禁近烬浸

0 1 2 3 4 5 6 7 8 9
300 尽劲荆兢茎睛晶鲸京
301 惊精粳经井警景颈静境
302 敬镜径痉靖竟竞净炯窘
303 揪究纠玖韭久灸九酒厩
304 救旧臼舅咎就疚鞠拘狙
305 疽居驹菊局咀矩举沮聚
306 拒据巨具距踞锯俱句惧
307 炬剧捐鹃娟倦眷卷绢撅
308 攫抉掘倔爵觉决诀绝均
309 菌钧军君峻

0 1 2 3 4 5 6 7 8 9
310 俊竣浚郡骏喀咖卡咯
311 开揩楷凯慨刊堪勘坎砍
312 看康慷糠扛抗亢炕考拷
313 烤靠坷苛柯棵磕颗科壳
314 咳可渴克刻客课肯啃垦
315 恳坑吭空恐孔控抠口扣
316 寇枯哭窟苦酷库裤夸垮
317 挎跨胯块筷侩快宽款匡
318 筐狂框矿眶旷况亏盔岿
319 窥葵奎魁傀

0 1 2 3 4 5 6 7 8 9
320 馈愧溃坤昆捆困括扩
321 廓阔垃拉喇蜡腊辣啦莱
322 来赖蓝婪栏拦篮阑兰澜
323 谰揽览懒缆烂滥琅榔狼
324 廊郎朗浪捞劳牢老佬姥
325 酪烙涝勒乐雷镭蕾磊累
326 儡垒擂肋类泪棱楞冷厘
327 梨犁黎篱狸离漓理李里
328 鲤礼莉荔吏栗丽厉励砾
329 历利傈例俐

0 1 2 3 4 5 6 7 8 9
330 痢立粒沥隶力璃哩俩
331 联莲连镰廉怜涟帘敛脸
332 链恋炼练粮凉梁粱良两
333 辆量晾亮谅撩聊僚疗燎
334 寥辽潦了撂镣廖料列裂
335 烈劣猎琳林磷霖临邻鳞
336 淋凛赁吝拎玲菱零龄铃
337 伶羚凌灵陵岭领另令溜
338 琉榴硫馏留刘瘤流柳六
339 龙聋咙笼窿

0 1 2 3 4 5 6 7 8 9
340 隆垄拢陇楼娄搂篓漏
341 陋芦卢颅庐炉掳卤虏鲁
342 麓碌露路赂鹿潞禄录陆
343 戮驴吕铝侣旅履屡缕虑
344 氯律率滤绿峦挛孪滦卵
345 乱掠略抡轮伦仑沦纶论
346 萝螺罗逻锣箩骡裸落洛
347 骆络妈麻玛码蚂马骂嘛
348 吗埋买麦卖迈脉瞒馒蛮
349 满蔓曼慢漫

0 1 2 3 4 5 6 7 8 9
350 谩芒茫盲氓忙莽猫茅
351 锚毛矛铆卯茂冒帽貌贸
352 么玫枚梅酶霉煤没眉媒
353 镁每美昧寐妹媚门闷们
354 萌蒙檬盟锰猛梦孟眯醚
355 靡糜迷谜弥米秘觅泌蜜
356 密幂棉眠绵冕免勉娩缅
357 面苗描瞄藐秒渺庙妙蔑
358 灭民抿皿敏悯闽明螟鸣
359 铭名命谬摸

国标一级汉字表

	0	1	2	3	4	5	6	7	8	9
360		摹	蘑	模	膜	磨	摩	魔	抹	末
361	莫	墨	默	沫	漠	寞	陌	谋	牟	某
362	拇	牡	亩	姆	母	墓	暮	幕	募	慕
363	木	目	睦	牧	穆	拿	哪	呐	钠	那
364	娜	纳	氖	乃	奶	耐	奈	南	男	难
365	囊	挠	脑	恼	闹	淖	呢	馁	内	嫩
366	能	妮	霓	倪	泥	尼	拟	你	匿	腻
367	逆	溺	蔫	拈	年	碾	撵	捻	念	娘
368	酿	鸟	尿	捏	聂	孽	啮	镊	镍	涅
369	您	柠	狞	凝	宁					

	0	1	2	3	4	5	6	7	8	9
370		拧	泞	牛	扭	钮	纽	脓	浓	农
371	弄	奴	努	怒	女	暖	虐	疟	挪	懦
372	糯	诺	哦	欧	鸥	殴	藕	呕	偶	沤
373	啪	趴	爬	帕	怕	琶	拍	排	牌	徘
374	湃	派	攀	潘	盘	磐	盼	畔	判	叛
375	乓	庞	旁	耪	胖	抛	咆	刨	炮	袍
376	跑	泡	呸	胚	培	裴	赔	陪	配	佩
377	沛	喷	盆	砰	抨	烹	澎	彭	蓬	棚
378	硼	篷	膨	朋	鹏	捧	碰	坯	砒	霹
379	批	披	劈	琵	毗					

	0	1	2	3	4	5	6	7	8	9
380		啤	脾	疲	皮	匹	痞	僻	屁	譬
381	篇	偏	片	骗	飘	漂	瓢	票	撇	瞥
382	拼	频	贫	品	聘	乒	坪	苹	萍	平
383	凭	瓶	评	屏	坡	泼	颇	婆	破	魄
384	迫	粕	剖	扑	铺	仆	莆	葡	菩	蒲
385	埔	朴	圃	普	浦	谱	曝	瀑	期	欺
386	栖	戚	妻	七	凄	漆	柒	沏	其	棋
387	奇	歧	畦	崎	脐	齐	旗	祈	祁	骑
388	起	岂	乞	企	启	契	砌	器	气	迄
389	弃	汽	泣	讫	掐					

	0	1	2	3	4	5	6	7	8	9
390		恰	洽	牵	扦	钎	铅	千	迁	签
391	仟	谦	乾	黔	钱	钳	前	潜	遣	浅
392	谴	堑	嵌	欠	歉	枪	呛	腔	羌	墙
393	蔷	强	抢	橇	锹	敲	悄	桥	瞧	乔
394	侨	巧	鞘	撬	翘	峭	俏	窍	切	茄
395	且	怯	窃	钦	侵	亲	秦	琴	勤	芹
396	擒	禽	寝	沁	青	轻	氢	倾	卿	清
397	擎	晴	氰	情	顷	请	庆	琼	穷	秋
398	丘	邱	球	求	囚	酋	泅	趋	区	蛆
399	曲	躯	屈	驱	渠					

	0	1	2	3	4	5	6	7	8	9
400		取	娶	龋	趣	去	圈	颧	权	醛
401	泉	全	痊	拳	犬	券	劝	缺	炔	瘸
402	却	鹊	榷	确	雀	裙	群	然	燃	冉
403	染	瓤	壤	攘	嚷	让	饶	扰	绕	惹
404	热	壬	仁	人	忍	韧	任	认	刃	妊
405	纫	扔	仍	日	戎	茸	蓉	荣	融	熔
406	溶	容	绒	冗	揉	柔	肉	茹	蠕	儒
407	孺	如	辱	乳	汝	入	褥	软	阮	蕊
408	瑞	锐	闰	润	若	弱	撒	洒	萨	腮
409	鳃	塞	赛	三	叁					

	0	1	2	3	4	5	6	7	8	9
410		伞	散	桑	嗓	丧	搔	骚	扫	嫂
411	瑟	色	涩	森	僧	莎	砂	杀	刹	沙
412	纱	傻	啥	煞	筛	晒	珊	苫	杉	山
413	删	煽	衫	闪	陕	擅	赡	膳	善	汕
414	扇	缮	墒	伤	商	赏	晌	上	尚	裳
415	梢	捎	稍	烧	芍	勺	韶	少	哨	邵
416	绍	奢	赊	蛇	舌	舍	赦	摄	射	慑
417	涉	社	设	砷	申	呻	伸	身	深	娠
418	绅	神	沈	审	婶	甚	肾	慎	渗	声
419	生	甥	牲	升	绳					

	0	1	2	3	4	5	6	7	8	9
420		省	盛	剩	胜	圣	师	失	狮	施
421	湿	诗	尸	虱	十	石	拾	时	什	食
422	蚀	实	识	史	矢	使	屎	驶	始	式
423	示	士	世	柿	事	拭	誓	逝	势	是
424	嗜	噬	适	仕	侍	释	饰	氏	市	恃
425	室	视	试	收	手	首	守	寿	授	售
426	受	瘦	兽	蔬	枢	梳	殊	抒	输	叔
427	舒	淑	疏	书	赎	孰	熟	薯	暑	曙
428	署	蜀	黍	鼠	属	术	述	树	束	戍
429	竖	墅	庶	数	漱					

	0	1	2	3	4	5	6	7	8	9
430		恕	刷	耍	摔	衰	甩	帅	栓	拴
431	霜	双	爽	谁	水	睡	税	吮	瞬	顺
432	舜	说	硕	朔	烁	斯	撕	嘶	思	私
433	司	丝	死	肆	寺	嗣	四	伺	似	饲
434	巳	松	耸	怂	颂	送	宋	讼	诵	搜
435	艘	擞	嗽	苏	酥	俗	素	速	粟	僳
436	塑	溯	宿	诉	肃	酸	蒜	算	虽	隋
437	随	绥	髓	碎	岁	穗	遂	隧	祟	孙
438	损	笋	蓑	梭	唆	缩	琐	索	锁	所
439	塌	他	它	她	塔					

	0	1	2	3	4	5	6	7	8	9
440		獭	挞	蹋	踏	胎	苔	抬	台	泰
441	酞	太	态	汰	坍	摊	贪	瘫	滩	坛
442	檀	痰	潭	谭	谈	坦	毯	袒	碳	探
443	叹	炭	汤	塘	搪	堂	棠	膛	唐	糖
444	倘	躺	淌	趟	烫	掏	涛	滔	绦	萄
445	桃	逃	淘	陶	讨	套	特	藤	腾	疼
446	誊	梯	剔	踢	锑	提	题	蹄	啼	体
447	替	嚏	惕	涕	剃	屉	天	添	填	田
448	甜	恬	舔	腆	挑	条	迢	眺	跳	贴
449	铁	帖	厅	听	烃					

	0	1	2	3	4	5	6	7	8	9
450		汀	廷	停	亭	庭	挺	艇	通	桐
451	酮	瞳	同	铜	彤	童	桶	捅	筒	统
452	痛	偷	投	头	透	凸	秃	突	图	徒
453	途	涂	屠	土	吐	兔	湍	团	推	颓
454	腿	蜕	褪	退	吞	屯	臀	拖	托	脱
455	鸵	陀	驮	驼	椭	妥	拓	唾	挖	哇
456	蛙	洼	娃	瓦	袜	歪	外	豌	弯	湾
457	玩	顽	丸	烷	完	碗	挽	晚	皖	惋
458	宛	婉	万	腕	汪	王	亡	枉	网	往
459	旺	望	忘	妄	威					

	0	1	2	3	4	5	6	7	8	9
460		巍	微	危	韦	违	桅	围	唯	惟
461	为	潍	维	苇	萎	委	伟	伪	尾	纬
462	未	蔚	味	畏	胃	喂	魏	位	渭	谓
463	尉	慰	卫	瘟	温	蚊	文	闻	纹	吻
464	稳	紊	问	嗡	翁	瓮	挝	蜗	涡	窝
465	我	斡	卧	握	沃	巫	呜	钨	乌	污
466	诬	屋	无	芜	梧	吾	吴	毋	武	五
467	捂	午	舞	伍	侮	坞	戊	雾	晤	物
468	勿	务	悟	误	昔	熙	析	西	硒	矽
469	晰	嘻	吸	锡	牺					

	0	1	2	3	4	5	6	7	8	9
470		稀	息	希	悉	膝	夕	惜	熄	烯
471	溪	汐	犀	檄	袭	席	习	媳	喜	铣
472	洗	系	隙	戏	细	瞎	虾	匣	霞	辖
473	暇	峡	侠	狭	下	厦	夏	吓	掀	锨
474	先	仙	鲜	纤	咸	贤	衔	舷	闲	涎
475	弦	嫌	显	险	现	献	县	腺	馅	羡
476	宪	陷	限	线	相	厢	镶	香	箱	襄
477	湘	乡	翔	祥	详	想	响	享	项	巷
478	橡	像	向	象	萧	硝	霄	削	哮	嚣
479	销	消	宵	淆	晓					

	0	1	2	3	4	5	6	7	8	9
480		小	孝	校	肖	啸	笑	效	楔	些
481	歇	蝎	鞋	协	挟	携	邪	斜	胁	谐
482	写	械	卸	蟹	懈	泄	泻	谢	屑	薪
483	芯	锌	欣	辛	新	忻	心	信	衅	星
484	腥	猩	惺	兴	刑	型	形	邢	行	醒
485	幸	杏	性	姓	兄	凶	胸	匈	汹	雄
486	熊	休	修	羞	朽	嗅	锈	秀	袖	绣
487	墟	戌	需	虚	嘘	须	徐	许	蓄	酗
488	叙	旭	序	畜	恤	絮	婿	绪	续	轩
489	喧	宣	悬	旋	玄					

	0	1	2	3	4	5	6	7	8	9
490		选	癣	眩	绚	靴	薛	学	穴	雪
491	血	勋	熏	循	旬	询	寻	驯	巡	殉
492	汛	训	讯	逊	迅	压	押	鸦	鸭	呀
493	丫	芽	牙	蚜	崖	衙	涯	雅	哑	亚
494	讶	焉	咽	阉	烟	淹	盐	严	研	蜒
495	岩	延	言	颜	阎	炎	沿	奄	掩	眼
496	衍	演	艳	堰	燕	厌	砚	雁	唁	彦
497	焰	宴	谚	验	殃	央	鸯	秧	杨	扬
498	佯	疡	羊	洋	阳	氧	仰	痒	养	样
499	漾	邀	腰	妖	瑶					

	0	1	2	3	4	5	6	7	8	9
500		摇	尧	遥	窑	谣	姚	咬	舀	药
501	要	耀	椰	噎	耶	爷	野	冶	也	页
502	掖	业	叶	曳	腋	夜	液	一	壹	医
503	揖	铱	依	伊	衣	颐	夷	遗	移	仪
504	胰	疑	沂	宜	姨	彝	椅	蚁	倚	已
505	乙	矣	以	艺	抑	易	邑	屹	亿	役
506	臆	逸	肄	疫	亦	裔	意	毅	忆	义
507	益	溢	诣	议	谊	译	异	翼	翌	绎
508	茵	荫	因	殷	音	阴	姻	吟	银	淫
509	寅	饮	尹	引	隐					

	0	1	2	3	4	5	6	7	8	9
510		印	英	樱	婴	鹰	应	缨	莹	萤
511	营	荧	蝇	迎	赢	盈	影	颖	硬	映
512	哟	拥	佣	臃	痈	庸	雍	踊	蛹	咏
513	泳	涌	永	恿	勇	用	幽	优	悠	忧
514	尤	由	邮	铀	犹	油	游	酉	有	友
515	右	佑	釉	诱	又	幼	迂	淤	于	盂
516	榆	虞	愚	舆	余	俞	逾	鱼	愉	渝
517	渔	隅	予	娱	雨	与	屿	禹	宇	语
518	羽	玉	域	芋	郁	吁	遇	喻	峪	御
519	愈	欲	狱	育	誉					

	0	1	2	3	4	5	6	7	8	9
520		浴	寓	裕	预	豫	驭	鸳	渊	冤
521	元	垣	袁	原	援	辕	园	员	圆	猿
522	源	缘	远	苑	愿	怨	院	曰	约	越
523	跃	钥	岳	粤	月	悦	阅	耘	云	郧
524	匀	陨	允	运	蕴	酝	晕	韵	孕	匝
525	砸	杂	栽	哉	灾	宰	载	再	在	咱
526	攒	暂	赞	赃	脏	葬	遭	糟	凿	藻
527	枣	早	澡	蚤	躁	噪	造	皂	灶	燥
528	责	择	则	泽	贼	怎	增	憎	曾	赠
529	扎	喳	渣	札	轧					

	0	1	2	3	4	5	6	7	8	9
530		铡	闸	眨	栅	榨	咋	乍	炸	诈
531	摘	斋	宅	窄	债	寨	瞻	毡	詹	粘
532	沾	盏	斩	辗	崭	展	蘸	栈	占	战
533	站	湛	绽	樟	章	彰	漳	张	掌	涨
534	杖	丈	帐	账	仗	胀	瘴	障	招	昭
535	找	沼	赵	照	罩	兆	肇	召	遮	折
536	哲	蛰	辙	者	锗	蔗	这	浙	珍	斟
537	真	甄	砧	臻	贞	针	侦	枕	疹	诊
538	震	振	镇	阵	蒸	挣	睁	征	狰	争
539	怔	整	拯	正	政					

	0	1	2	3	4	5	6	7	8	9
540		帧	症	郑	证	芝	枝	支	吱	蜘
541	知	肢	脂	汁	之	织	职	直	植	殖
542	执	值	侄	址	指	止	趾	只	旨	纸
543	志	挚	掷	至	致	置	帜	峙	制	智
544	秩	稚	质	炙	痔	滞	治	窒	中	盅
545	忠	钟	衷	终	种	肿	重	仲	众	舟
546	周	州	洲	诌	粥	轴	肘	帚	咒	皱
547	宙	昼	骤	珠	株	蛛	朱	猪	诸	诛
548	逐	竹	烛	煮	拄	瞩	嘱	主	著	柱
549	助	蛀	贮	铸	筑					

	0	1	2	3	4	5	6	7	8	9
550		住	注	祝	驻	抓	爪	拽	专	砖
551	转	撰	赚	篆	桩	庄	装	妆	撞	壮
552	状	椎	锥	追	赘	坠	缀	谆	准	捉
553	拙	卓	桌	琢	茁	酌	啄	着	灼	浊
554	兹	咨	资	姿	滋	淄	孜	紫	仔	籽
555	滓	子	自	渍	字	鬃	棕	踪	宗	综
556	总	纵	邹	走	奏	揍	租	足	卒	族
557	祖	诅	阻	组	钻	纂	嘴	醉	最	罪
558	尊	遵	昨	左	佐	柞	做	作	坐	座
559										

国标二级汉字表

0 1 2 3 4 5 6 7 8 9
560 亍丌兀丐廿卅丕亘丞
561 鬲孬噩丨禺丿匕乇夭爻
562 卮氐囟胤馗毓睾鼗丶亟
563 鼐乜乩亓芈孛啬嘏仄厍
564 厝厣厥厮靥赝匚叵匦匮
565 匾赜卦卣刂刈刎刭刳刿
566 剀剌剞剡剜蒯剽劂劁劐
567 劓冂罔亻仃仉仂仨仡仫
568 仞伛仳伢佤仵伥伧伉伫
569 佞佧攸佚佝

0 1 2 3 4 5 6 7 8 9
570 佟佗伲伽佶佴侑侉侃
571 侏佾佻侪佼侬侔俦俨俪
572 俅俚俣俜俑俟俸倩偌俳
573 倬倏倮倭俾倜倌倥倨偾
574 偃偕偈偎偬偻傥傧傩傺
575 僖儆僭僬僦僮儇儋仝氽
576 佘佥俎龠汆籴兮巽黉馘
577 冁夔勹匍訇匐凫夙兕亠
578 兖亳衮袤亵脔裒禀嬴蠃
579 羸冫冱冽冼

0 1 2 3 4 5 6 7 8 9
580 凇冖冢冥讠讦讧讪讴
581 讵讷诂诃诋诏诎诒诓诔
582 诖诘诙诜诟诠诤诨诩诮
583 诰诳诶诹诼诿谀谂谄谇
584 谌谏谑谒谔谕谖谙谛谘
585 谝谟谠谡谥谧谪谫谮谯
586 谲谳谵谶卩卺阝阢阡阱
587 阪阽阼陂陉陔陟陧陬陲
588 陴隈隍隗隰邗邛邝邙邬
589 邡邴邳邶邺

0 1 2 3 4 5 6 7 8 9
590 邸邰郏郅邾郐郄郇郓
591 郦郢郜郗郛郫郯郾鄄鄢
592 鄞鄣鄱鄯鄹酃酆刍奂劢
593 劬劭劾哿勐勖勰叟燮矍
594 廴凵凼鬯厶弁畚巯坌垩
595 垡塾墼壅壑圩圬圪圳圹
596 圮圯坜圻坂坩垅坫垆坼
597 坻坨坭坶坳垭垤垌垲埏
598 垧垴垓垠埕埘埚埙埒垸
599 埴埯埸埤埝

0 1 2 3 4 5 6 7 8 9
600 堋堍埽埭堀堞堙塄堠
601 塥塬墁墉墚墀馨鼙懿艹
602 艽艿芏芊芨芄芎芑芗芙
603 芫芸芾芰苈苊苣芘芷芮
604 苋苌苁芩芴芡芪芟苄苎
605 芤苡茉苷苤茏茇苜苴苒
606 苘茌苻苓茑茚茆茔茕苠
607 苕茜荑荛荜茈莒茼茴茱
608 莛荞茯荏荇荃荟荀茗荠
609 茭茺茳荦荥

0 1 2 3 4 5 6 7 8 9
610 荨茛荩荬荪荭荮莰荸
611 莳莴莠莪莓莜莅荼莶莩
612 荽莸荻莘莞莨莺莼菁萁
613 菥菘堇萘萋菝菽菖萜萸
614 萑萆菔菟萏萃菸菹菪菅
615 菀萦菰菡葜葑葚葙葳蒇
616 蒈葺蒉葸萼葆葩葶蒌蒎
617 萱葭蓁蓍蓐蓦蒽蓓蓊蒿
618 蒺蓠蒡蒹蒴蒗蓥蓣蔌甍
619 蔸蓰蔹蔟蔺

0 1 2 3 4 5 6 7 8 9
620 蕖蔻蓿蓼蕙蕈蕨蕤蕞
621 蕺瞢蕃蕲蕻薤薨薇薏蕹
622 薮薜薅薹薷薰藓藁藜藿
623 蘧蘅蘩蘖蘼廾弈夼奁耷
624 奕奚奘匏尢尥尬尴扌扪
625 抟抻拊拚拗拮挢拶挹捋
626 捃掭揶捱捺掎掴捭掬掊
627 捩掮掼揲揸揠揿揄揞揎
628 摒揆掾摅摁搋搛搠搌搦
629 搡摞撄摭撖

0 1 2 3 4 5 6 7 8 9
630 摺撷撸撙撺擀擐擗擤
631 擢攉攥攮弋忒甙弑卟叱
632 叽叩叨叻吒吖吆呋呒呓
633 呔呖呃吡呗呙吣吲咂咔
634 呷呱呤咚咛咄呶呦咝哐
635 咭哂咴哒咧咦哓哔呲咣
636 哕咻咿哌哙哚哜咩咪咤
637 哝哏哞唛哧唠哽唔哳唢
638 唣唏唑唧唪啧喏喵啉啭
639 啁啕唿啐唼

0 1 2 3 4 5 6 7 8 9
640 唷啖啵啶啷唳唰啜喋
641 嗒喃喱喹喈喁喟啾嗖喑
642 啻嗟喽喾喔喙嗪嗷嗉嘟
643 嗑嗫嗬嗔嗦嗝嗄嗯嗥嗲
644 嗳嗌嗍嗨嗵嗤辔嘞嘈嘌
645 嘁嘤嘣嗾嘀嘧嘭噘嘹噗
646 嘬噍噢噙噜噌噔嚆噤噱
647 噫噻噼嚅嚓嚯囔囗囝囡
648 囵囫囹囿圄圊圉圜帏帙
649 帔帑帱帻帼

0 1 2 3 4 5 6 7 8 9
650 帷幄幔幛幞幡岌屺岍
651 岐岖岈岘岙岑岚岜岵岢
652 岽岬岫岱岣峁岷峄峒峤
653 峋峥崂崃崧崦崮崤崞崆
654 崛嵘崾崴崽嵬嵛嵯嵝嵫
655 嵋嵊嵩嵴嶂嶙嶝豳嶷巅
656 彳彷徂徇徉後徕徙徜徨
657 徭徵徼衢彡犭犰犴犷犸
658 狃狁狎狍狒狨狯狩狲狴
659 狷猁狳猃狺

0 1 2 3 4 5 6 7 8 9
660 狻猗猓猡猊猞猝猕猢
661 猹猥猬猸猱獐獍獗獠獬
662 獯獾舛夥飧夤夂饣饧饨
663 饩饪饫饬饴饷饽馀馄馇
664 馊馍馐馑馓馔馕庀庑庋
665 庖庥庠庹庵庾庳赓廒廑
666 廛廨廪膺忄忉忖忏怃忮
667 怄忡忤忾怅怆忪忭忸怙
668 怵怦怛怏怍怩怫怊怿怡
669 恸恹恻恺恂

0 1 2 3 4 5 6 7 8 9
670 恪恽悖悚悭悝悃悒悌
671 悛惬悻悱惝惘惆惚悴愠
672 愦愕愣惴愀愎愫慊慵憬
673 憔憧憷懔懵忝隳闩闫闱
674 闳闵闶闼闾阃阄阆阈阊
675 阋阌阍阏阒阕阖阗阙阚
676 丬爿戕氵汔汜汊沣沅沐
677 沔沌汨汩汴汶沆沩泐泔
678 沭泷泸泱泗沲泠泖泺泫
679 泮沱泓泯泾

0 1 2 3 4 5 6 7 8 9
680 洹洧洌浃浈洇洄洙洎
681 洫浍洮洵洚浏浒浔洳涑
682 浯涞涠浞涓涔浜浠浼浣
683 渚淇淅淞渎涿淠渑淦淝
684 淙渖涫渌涮渫湮湎湫溲
685 湟溆湓湔渲渥湄滟溱溘
686 滠漭滢溥溧溽溻溷滗溴
687 滏溏滂溟潢潆潇漤漕滹
688 漯漶潋潴漪漉漩澉澍澌
689 潸潲潼潺濑

0 1 2 3 4 5 6 7 8 9
690 濉澧澹澶濂濡濮濞濠
691 濯瀚瀣瀛瀹瀵灏灞宀宄
692 宕宓宥宸甯骞搴寤寮褰
693 寰蹇謇辶迓迕迥迮迤迩
694 迦迳迨逅逄逋逦逑逍逖
695 逡逵逶逭逯遄遑遒遐遨
696 遘遢遛暹遴遽邂邈邃邋
697 彐彗彖彘尻咫屐屙孱屣
698 屦羼弪弩弭艴弼鬻屮妁
699 妃妍妩妪妣

0 1 2 3 4 5 6 7 8 9
700 妗姊妫妞妤姒妲妯姗
701 妾娅娆姝娈姣姘姹娌娉
702 娲娴娑娣娓婀婧婊婕娼
703 婢婵胬媪媛婷婺媾嫫媲
704 嫒嫔媸嫠嫣嫱嫖嫦嫘嫜
705 嬉嬗嬖嬲嬷孀尕尜孚孥
706 孳孑孓孢驵驷驸驺驿驽
707 骀骁骅骈骊骐骒骓骖骘
708 骛骜骝骟骠骢骣骥骧纟
709 纡纣纥纨纩

0 1 2 3 4 5 6 7 8 9
710 纭纰纾绀绁绂绉绋绌
711 绐绔绗绛绠绡绨绫绮绯
712 绱绲缍绶绺绻绾缁缂缃
713 缇缈缋缌缏缑缒缗缙缜
714 缛缟缡缢缣缤缥缦缧缪
715 缫缬缭缯缰缱缲缳缵幺
716 畿巛甾邕玎玑玮玢玟珏
717 珂珑玷玳珀珉珈珥珙顼
718 琊珩珧珞玺珲琏琪瑛琦
719 琥琨琰琮琬

国标二级汉字表

0 1 2 3 4 5 6 7 8 9
720 琛琚瑁瑜瑗瑕瑙瑷瑭
721 瑾璜璎璀璁璇璋璞璨璩
722 璐璧瓒璺韪韫韬杌杓杞
723 杈杩枥枇杪杳枘枧杵枨
724 枞枭枋杷杼柰栉柘栊柩
725 枰栌柙枵柚枳柝栀柃枸
726 柢栎柁柽栲栳桠桡桎桢
727 桄桤梃栝桕桦桁桧桀栾
728 桊桉栩梵梏桴桷梓桫棂
729 楮棼椟椠棹

0 1 2 3 4 5 6 7 8 9
730 椤棰椋椁楗棣椐楱椹
731 楠楂楝榄楫榀榘楸椴槌
732 榇榈槎榉楦楣楹榛榧榻
733 榫榭槔榱槁槊槟榕槠榍
734 槿樯槭樗樘橥槲橄樾檠
735 橐橛樵檎橹樽樨橘橼檑
736 檐檩檗檫猷獒殁殂殇殄
737 殒殓殍殚殛殡殪轫轭轱
738 轲轳轵轶轸轷轹轺轼轾
739 辁辂辄辇辋

0 1 2 3 4 5 6 7 8 9
740 辍辎辏辘辚軎戋戗戛
741 戟戢戡戥戤戬臧瓯瓴瓿
742 甏甑甓攴旮旯旰昊昙杲
743 昃昕昀炅曷昝昴昱昶昵
744 耆晟晔晁晏晖晡晗晷暄
745 暌暧暝暾曛曜曦曩贲贳
746 贶贻贽赀赅赆赈赉赇赍
747 赕赙觇觊觋觌觎觏觐觑
748 牮犟牝牦牯牾牿犄犋犍
749 犏犒挈挲掰

0 1 2 3 4 5 6 7 8 9
750 搿擘耄毪毳毽毵毹氅
751 氇氆氍氕氘氙氚氡氩氤
752 氪氲攵敕敫牍牒牖爰虢
753 刖肟肜肓肼朊肽肱肫肭
754 肴肷胧胨胩胪胛胂胄胙
755 胍胗朐胝胫胱胴胭脍脎
756 胲胼朕脒豚脶脞脬脘脲
757 腈腌腓腴腙腚腱腠腩腼
758 腽腭腧塍媵膈膂膑滕膣
759 膪臌朦臊膻

0 1 2 3 4 5 6 7 8 9
760 臁膦欤欷欹歃歆歙飑
761 飒飓飕飙飚殳彀毂觳斐
762 齑斓於旆旄旃旌旎旒旖
763 炀炜炖炝炻烀炷炫炱烨
764 烊焐焓焖焯焱煳煜煨煅
765 煲煊煸煺熘熳熵熨熠燠
766 燔燧燹爝爨灬焘煦熹戾
767 戽扃扈扉礻祀祆祉祛祜
768 祓祚祢祗祠祯祧祺禅禊
769 禚禧禳忑忐

0 1 2 3 4 5 6 7 8 9
770 怼恝恚恧恁恙恣悫愆
771 愍慝憩憝懋懑戆肀聿沓
772 泶淼矶矸砀砉砗砘砑斫
773 砭砜砝砹砺砻砟砼砥砬
774 砣砩硎硭硖硗砦硐硇硌
775 硪碛碓碚碇碜碡碣碲碹
776 碥磔磙磉磬磲礅磴礓礤
777 礞礴龛黹黻黼盱眄眍盹
778 眇眈眚眢眙眭眦眵眸睐
779 睑睇睃睚睨

0 1 2 3 4 5 6 7 8 9
780 睢睥睿瞍睽瞀瞌瞑瞟
781 瞠瞰瞵瞽町畀畎畋畈畛
782 畲畹疃罘罡罟詈罨罴罱
783 罹羁罾盍盥蠲钅钆钇钋
784 钊钌钍钏钐钔钗钕钚钛
785 钜钣钤钫钪钭钬钯钰钲
786 钴钶钷钸钹钺钼钽钿铄
787 铈铉铊铋铌铍铎铐铑铒
788 铕铖铗铙铘铛铞铟铠铢
789 铤铥铧铨铪

0 1 2 3 4 5 6 7 8 9
790 铩铫铮铯铳铴铵铷铹
791 铼铽铿锃锂锆锇锉锊锍
792 锎锏锒锓锔锕锖锘锛锝
793 锞锟锢锪锫锩锬锱锲锴
794 锶锷锸锼锾锿镂锵镄镅
795 镆镉镌镎镏镒镓镔镖镗
796 镘镙镛镞镟镝镡镢镤镥
797 镦镧镨镩镪镫镬镯镱镲
798 镳锺矧矬雉秕秭秣秫稆
799 嵇稃稂稞稔

0 1 2 3 4 5 6 7 8 9
800 稹稷穑黏馥穰皈皎皓
801 皙皤瓞瓠甬鸠鸢鸨鸩鸪
802 鸫鸬鸲鸱鸶鸸鸷鸹鸺鸾
803 鹁鹂鹄鹆鹇鹈鹉鹋鹌鹎
804 鹑鹕鹗鹚鹛鹜鹞鹣鹦鹧
805 鹨鹩鹪鹫鹬鹱鹭鹳疒疔
806 疖疠疝疬疣疳疴疸痄疱
807 疰痃痂痖痍痣痨痦痤痫
808 痧瘃痱痼痿瘐瘀瘅瘌瘗
809 瘊瘥瘘瘕瘙

0 1 2 3 4 5 6 7 8 9
810 瘛瘼瘢瘠癀瘭瘰瘿瘵
811 癃瘾瘳癍癞癔癜癖癫癯
812 翊竦穸穹窀窆窈窕窦窠
813 窬窨窭窳衤衩衲衽衿袂
814 袢裆袷袼裉裢裎裣裥裱
815 褚裼裨裾裰褡褙褓褛褊
816 褴褫褶襁襦襻疋胥皲皴
817 矜耒耔耖耜耠耢耥耦耧
818 耩耨耱耋耵聃聆聍聒聩
819 聱覃顸颀颃

0 1 2 3 4 5 6 7 8 9
820 颉颌颍颏颔颚颛颞颟
821 颡颢颥颦虍虔虬虮虿虺
822 虼虻蚨蚍蚋蚬蚝蚧蚣蚪
823 蚓蚩蚶蛄蚵蛎蚰蚺蚱蚯
824 蛉蛏蚴蛩蛱蛲蛭蛳蛐蜓
825 蛞蛴蛟蛘蛑蜃蜇蛸蜈蜊
826 蜍蜉蜣蜻蜞蜥蜮蜚蜾蝈
827 蜴蜱蜩蜷蜿螂蜢蝽蝾蝻
828 蝠蝰蝌蝮螋蝓蝣蝼蝤蝙
8229 蝥螓螯螨蟒

0 1 2 3 4 5 6 7 8 9
830 蟆螈螅螭螗螃螫蟥螬
831 螵螳蟋蟓螽蟑蟀蟊蟛蟪
832 蟠蟮蠖蠓蟾蠊蠛蠡蠹蠼
833 缶罂罄罅舐竺竽笈笃笄
834 笕笊笫笏筇笸笪笙笮笱
835 笠笥笤笳笾笞筘筚筅筵
836 筌筝筠筮筻筢筲筱箐箦
837 箧箸箬箝箨箅箪箜箢箫
838 箴篑篁篌篝篚篥篦篪簌
839 篾篼簏簖簋

0 1 2 3 4 5 6 7 8 9
840 簟簪簦簸籁籀臾舁舂
841 舄臬衄舡舢舣舭舯舨舫
842 舸舻舳舴舾艄艉艋艏艚
843 艟艨衾袅袈裘裟襞羝羟
844 羧羯羰羲籼敉粑粝粜粞
845 粢粲粼粽糁糇糌糍糈糅
846 糗糨艮暨羿翎翕翥翡翦
847 翩翮翳糸絷綦綮繇纛麸
848 麴赳趄趔趑趱赧赭豇豉
849 酊酐酎酏酤

0 1 2 3 4 5 6 7 8 9
850 酢酡酰酩酯酽酾酲酴
851 酹醌醅醐醍醑醢醣醪醭
852 醮醯醵醴醺豕鹾趸跫踅
853 蹙蹩趵趿趼趺跄跖跗跚
854 跞跎跏跛跆跬跷跸跣跹
855 跻跤踉跽踔踝踟踬踮踣
856 踯踺蹀踹踵踽踱蹉蹁蹂
857 蹑蹒蹊蹰蹶蹼蹯蹴躅躏
858 躔躐躜躞豸貂貊貅貘貔
859 斛觖觞觚觜

0 1 2 3 4 5 6 7 8 9
860 觥觫觯訾謦靓雩雳雯
861 霆霁霈霏霎霪霭霰霾龀
862 龃龅龆龇龈龉龊龌黾鼋
863 鼍隹隼隽雎雒瞿雠銎銮
864 鋈錾鍪鏊鎏鐾鑫鱿鲂鲅
865 鲆鲇鲈稣鲋鲎鲐鲑鲒鲔
866 鲕鲚鲛鲞鲟鲠鲡鲢鲣鲥
867 鲦鲧鲨鲩鲫鲭鲮鲰鲱鲲
868 鲳鲴鲵鲶鲷鲺鲻鲼鲽鳄
869 鳅鳆鳇鳊鳋

0 1 2 3 4 5 6 7 8 9
870 鳌鳍鳎鳏鳐鳓鳔鳕鳗
871 鳘鳙鳜鳝鳟鳢靼鞅鞑鞒
872 鞔鞯鞫鞣鞲鞴骱骰骷鹘
873 骶骺骼髁髀髅髂髋髌髑
874 魅魃魇魉魈魍魑飨餍餮
875 饕饔髟髡髦髯髫髻髭髹
876 鬈鬏鬓鬟鬣麽麾縻麂麇
877 麈麋麒鏖麝麟黛黜黝黠
878 黟黢黩黧黥黪黯鼢鼬鼯
879 鼹鼷鼽鼾齄

FBASIC 语言

使用手册

F　BASIC 语言使用手册目录

一、预备知识

1.1 大有可为的新天地

近年来，各种各样的游戏机如潮水般涌入我国的千万个家庭，据统计，国内单纯拥有游戏机的家庭近三千万。游戏机操作简单，游戏节目引人入胜，不仅能启发智力，锻炼思维能力、反应能力及观察事物摸索规律的能力，还能向人传达诸如正必压邪、多劳多得、勇者胜等社会崇尚的主流思想。因此，游戏机越来越受到广大群众的喜爱。

游戏机在我国发展了十年，使用的各种游戏节目都是日本和美国的作品，而我国在编制游戏机用的电子游戏程序方面几乎还是一片空白。在日本，有大量介绍游戏程序的杂志和书籍，仅在北京图书馆就有三种专门介绍游戏程序的日文杂志。而时至今日，国内只有一本《家庭电脑与游戏机》杂志，怎么能满足几千万游戏迷?!

在日本，青少年是编制和购买游戏程序的主力军，而在我国，还没有编制游戏的高水平人才，所以这项事业大有可为!

1.2 F BASIC 简介

简单地说，BASIC 的意思是计算机初学者使用的语言，而 F BASIC 的意思是家庭用 BASIC。BASIC 是目前世界范围内应用最广，用户最多、最容易学习的一种计算机程序设计语言。掌握了它，就不难学习其它的计算机程序设计语言。

F BASIC 与其它 BASIC 的区别在于 F BASIC 提供了多种控制卡通人物和背景画面的命令，可以很方便地把卡通人物，比如玛丽奥、丽莎、小企鹅等，叠加在自己设计的场景中，比如千里雪原，大森林，迷宫城堡，他们可以有各种作为，比如飞奔、跳跃、攀登等。

F BASIC 为你带来的是极富挑战性的全新天地！已经有很多青少年朋友，甚至中年朋友给我寄来了优秀的游戏程序和应用程序，我们为他们发行了《程序集》，还有一些程序还刊登在国内有名的杂志上。现在是你大显身手的时候了，让我们一起来推动中国的电子游戏事业，做勇敢开拓的一代人!

1.3 怎样学习本手册

由于计算机语言对于普通用户来说比较陌生，有浓厚的神秘感，所以本手册的编写力图通俗易懂，适合自学，章节安排上遵循由浅入深，寓学习于娱乐的原则。

第一章入门须知的背景资料。

第二章介绍了计算板、打字板和音乐板的用法，最适合学龄前儿童和小学生学习和娱乐。

第三章是 F BASIC 入门，通过大量的程序例子和注解分析，介绍了 F BASIC 的大多数命令和概念。在本章中，注重讲解每条命令怎么用，而不过多地分析为什么要这样用，这样可使初学者避开复杂的计算和专业概念，从而提高学习兴趣。

第四章是语法篇，把上一章介绍的各条命令分类，并逐条剖析，进入 F BASIC 的更高境界，并可供读者经常查询命令的格式和功能之用。

第五章是应用篇，介绍了《程序范例》中收录的八个游戏程序。通过对这些程序的细致分析，可以引导读者提高游戏程序的设计能力。

最后是一个附录，提供了各种参考数据和图表。

学习F BASIC的最好方法是边看书边练习，只看书不练习，所学的知识得不到巩固。本手册始终贯彻“模仿”的思想，让各种文化程序的人通过模仿而达到学习F BASIC的目的。文化水平的不同，只不过是理解的深度不同，进步快慢不同，编制出的游戏程序的复杂程度不同罢了。

编制游戏程序的好坏，取决于读者在模仿基础上的创造力，青少年通过编制优化程序，可大大丰富发展想象力和思维能力，最终促进学业进步。

二、初学者天地

2.1 进入F BASIC世界

超级学习卡采用全菜单提示，使用户一目了然，你只需按照提示按动相应的键，就能进入F BASIC世界。

2.2 片头提示

进入F BASIC后会显示如下：

```
F BASIC  V5.OC
YUXING  COMPUTER  CO.
4030  BYTES  FREE
OK
■
```

第一行表示这是5.OC版的F BASIC。

第二行表示制作单位是“裕兴电脑公司”。

第三行表示用户在F BASIC状态下有4030个字节的空间来存放程序。

OK表示一切正常。

光标在OK下面闪烁不停，等待你输入指令或程序。这时，你可以输入程序或是键入SYSTEM，然后按回车键进入一个分菜单，请参见2.6节的介绍。

2.3 计算板

图2.3是计算板的示意图，“CALCUATORBOARD”是计算板的英文名字。

羽毛笔羽毛光标：表示数据输入的位置。整个屏幕由8行组成，每行可以打入24个字符，F1代表“加”，F2代表“减”，F3代表“乘”，F4代表“除”，F5代表“左括号”，F6代表“右括弧”，F7代表“小数点”，F8代表等号。F8旁边是墨水瓶。使用时按下这些相应的功能键，就能得到相应的符号。

图2.3中的“CALCUATOR BOARD”的意思是计算板之意。

羽毛是光标，叫羽毛光标；表示数据输入的位置。整个屏幕由8行组成，每行可以打入24个字符。

[F1]代表“加”，[F2]代表“减”，[F3]代表“乘”，[F4]代表“除”，[F5]代表“左括弧”，[F6]代表“右

括弧”，F7 代表“小数点”，F8 代表等号。F8 旁边是墨水瓶。使用时按下功能键，便得到相应的符号。

图 2.4 是计算板用法示意。

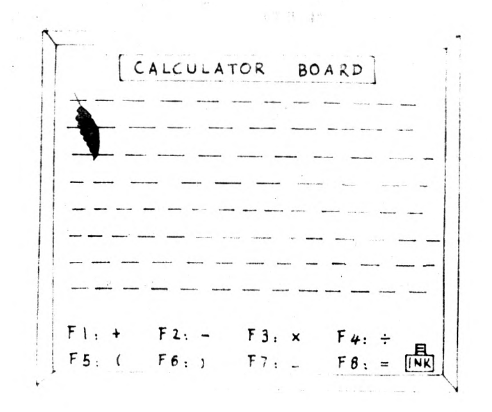


图 2.3　计算板示意

图 2.4　计算板用法示意

▲计算板运算规则

(1)算式

加减乘除四则运算。

括弧最多可用 10 次。

(2)有效数字

8 位，8 位之后的数字自动省略。

±(99999999～0.0000001)

(3)计算优先顺序

先括弧再乘除后加减。

▲计算板使用方法

图 2.4 中已经写入了各种数字，下面对应图 2.4 介绍计算器板的用法。

(1)表示一行内可做多道习题。

(2)表示算式的长度允许超过一行。

(3)表示负数可以用括弧括起来。

(4)表示算式前后至少有一个空格。

(5)表示也可输入字符，但不参加计算。

(6)表示算式中若插入了空格，空格前的内容不参加计算。

(7)如果运算结果在第 8 行显示不下，将显示在第 9 行。

▲下述用法不能成立

(1)　130YUAN×2=WRONG　　;未定义的符号或字符不能在算式中使用 WRONG 表示出错。

(2)　4=WRONG　　;不构成算式。

(3)　10+3　=WRONG　　;等号前插入了空格。

(4)　20000×300000=OVERFLOW　　;计算结果超过 8 位。OVERFLOW 表示结果超长。

(5)　ABC　　;当上一行最后一个字符是字母，下行计算出错。这

1+2=WRONG　　　　　　　　　时可删去 C,造成一个空格即可。

▲算式的修改方法

(1)用回车键可把羽毛光标移动到下一行最左边位置,即换行。

(2)用 ↑ ↓ → ← 方向键可以移动羽毛光标。

(3)用清除键 DEL 和插入键 INS 可以删去或增加算式中的符号。

(4)一修改算式,等号就自动删去,要计算时要再按 F8 。

(5)如果新的结果会叠在后面的算式上,这时被叠的算式部分将被删去。

(6)按 HOME 键,屏幕将被清除。

▲退出计算器板的方法

(1)按 ESC 键,可以退出计算板。

2.4 打字板

图 2.5 是打字板的示意图,“MESSBOARD”是打字板的英文名字。打字板的大小为 17 行,一行 24 个字符(24×17)。利用打字板可以写文章、作图、打字等等。

▲利用打字板练习打字。由于键盘上英文字母排列和普通打字机相同,故而可以用来练习打字。

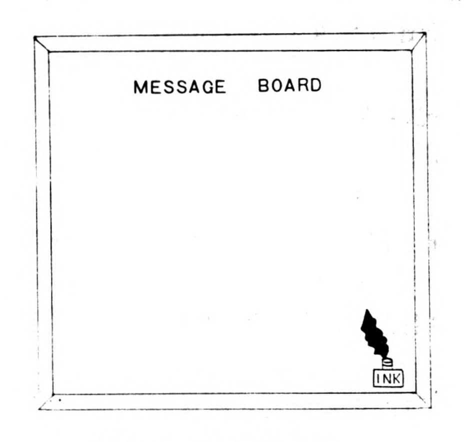


图 2.5　打字板示意

▲修改方法

(1)用回车键右把羽毛光标移到下一行最左边位置。

(2)用 ↑ ↓ → ← 键来移动羽毛光标到所需修改的位置。

(3)用 DEL 或 INS 键可以删除或增加字符。

(4)按下 HOME 键可以清屏幕。

▲退出打字板的方法

(1)　按 ESC 键可退出打字板。

2.5 音乐板

图 2.6 是音乐板的示意图,“MUSICBOARD”是音乐板的英文名字。音乐板有四大行,每一大行三小行,三小行对应地被称为上部、中部、下部。每一行可写入 24 个字符。在板的下方有些提示,含义为:

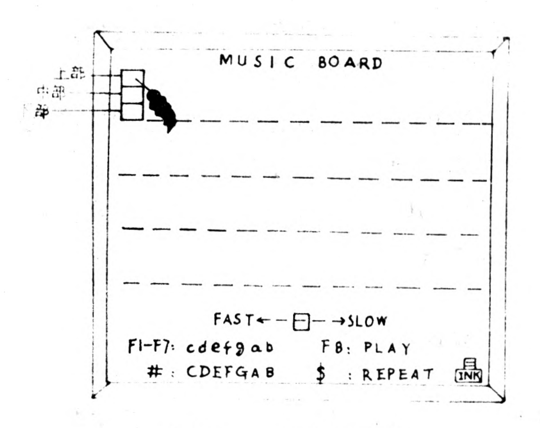


图 2.6 音乐板示意

(1) FAST<———Y———>SLOW

表示演奏速度。在演奏过程中用方向键→←可以改变。绿色光标向左移动时演奏速度加快，向右移动演奏速度变慢。

(2) F1—F7:cdefgab

按F1—F7功能键，对应于 cd……b。

cde…b 对应于“多来咪…都”。

(3) (#):CDEFGAB

CD…B 对应于“多来…都”。但比用小写的 cd…b 分别低半个音。

(4) F8:PLAY

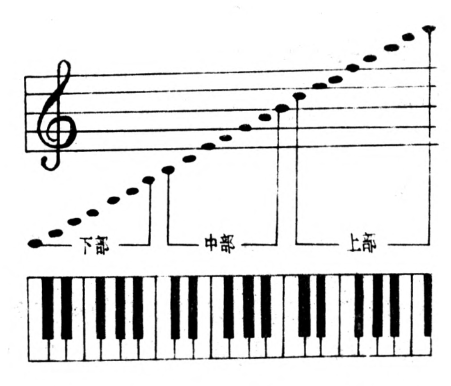
按F8开始演奏，羽毛光标被插入墨水瓶中。

(5) $:REPEAT

在乐谱中插入“$”，可以不停地重复演奏。

▲音乐板的使用方法说明

(1) 上部、中部、下部均可输入音符，并可同时演奏出三部合音。三部分别对应的 5 线谱如图 2.7 所示。



低音部　　中音部　　高音部

图 2.7

(2) 按[F1]—[F7]出现 cd…b 的音，按[SHIFT]加[F1]—[F7]将出现 CD…B 的音，图 2.8 是两者之间的关系示意。按功能键与直接从键盘上输入相应的字符效果相当。

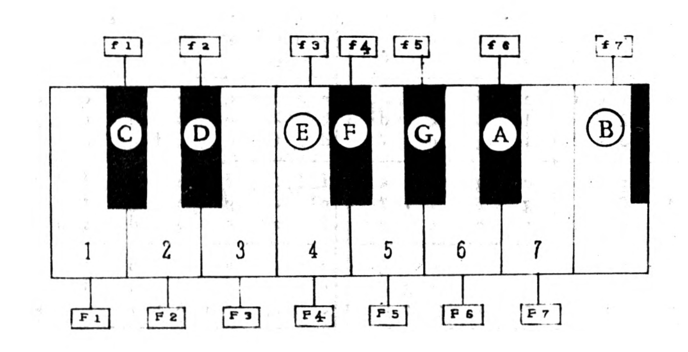


图 2.8

(3) 在上部输入数字 1、2、3、4、5、6、7，发高音域音，在中部输入数字 1、2、…7 发中音域的音，在下部输入数字 1、2、…7 可以发出低音部更低的音。

(4) 字符与数字一起可以表现 4 个 8 度音。

(5) 空格等未定义的字符可以作为休止符。

(6) "$"为重复演奏符。

(7) 在演奏过程中按[F8]键，可从头开始演奏。

(8) 按空格键可以终止演奏。

(9) 用[→][←]键改变演奏速度。

(10) 按[F8]键开始演奏。

▲音符的修改方法

(1) 用回车键可以把羽毛光标移动一大行。

(2) 用方向键[→][←][↑][↓]可以任意移动羽毛光标。

(3) 使用[DEL][INS]键可以删去或增加音符。

(4) 用[HOME]键可以清除屏幕。

▲退出音乐板的方法

(1) 按[ESC]键可退出音乐板。

▲乐谱示例

MUSIC BOARD

```
c   e   e   e   b c g             g f e   e   e
    1   1   1   b a 3             3 2 1   1   1
    c   c   c   5 4 c   5   5     5   c   c   c
-------------------------------------------------
d c g           g f e   f g f   e   d   7   5
b a 3           3 2 1   2 3 2   1   b   g   d
5 4 c   5   5   5   c   d e f   f   g   d   7
-------------------------------------------------
c   g   f e f   g   a   g       c   g   f e a
g   3   2 1 2   3   4   3       g   3   2 1 2
5   c   c   c   c   c   c   c   c   c   c   c
-------------------------------------------------
g   a   g       c   a   g       a   e   d   g
3   2   3       g   4   3               b g b
c   c   c   c   c   c   c   c   c   c   c c $
```

(1)

MUSIC BOARD

```
e e e   e e e   e g c d e     f f f   f e e
c   e   g   e   c   e   g   e   c   f   g   e
c   g   c   g               c c d e f   a   c   g
-------------------------------------------------
e d d e d   g   e e e   e e e   e g c d e
b   a   g   b   c   e   g   c   d   e   g   e
d   f   d   g f c   g   c   g   c   g   e c d e
-------------------------------------------------
f f f   f e e   g g f d c       5 e d c 5
c   f   g   e   g   f   c       c   e   g   e
f   a   c   g   d f g 5 c 5 6 7 c   g   c 5 5 5
-------------------------------------------------
5 e d c a       6 f e d g g g g a g f d c
c   e   f   d   a   f   b   g   f   d   c
c   g   f 6 6 6 f   a   g   b   g 5 6 7 C   5 $
```

(2)

MUSIC BOARD

```
5 c c c e e e g e a c c f f e d 6   d
a B g a e b g e b g g b e g f     e g     a B
c   f c e   c e c   e c d   c d   g d d f     f
-------------------------------------------------
                e e e g       f     d
                                  f e         g
f a B f a B f a e b g e b g e b d a f d       d
-------------------------------------------------
  f a   c   6 5 c   c c           c   7 6   7 c
  g d g       c   e g e   e f g a   f     f   g
g     g     g   c     c   c d e f   c   f   c d
-------------------------------------------------
7 7 5 3           6 6 5 6     c d c 6 5 c
  e     e f g g   f f e f   a   f     f   g
e   7   e   d e d   d   c   c   d   g   c   1 $
```

(3)

MUSIC BOARD

```
5       6 5 c d e       g   5   f   e   d   e c
c     c e c     c     c e   d     d   f d   d
c e g e c e g e c e g e c e g e   g   e   g   g
-------------------------------------------------
5       6       6       6     e f   e     d g d
a f   b   c f     d f a   d f d   d f a   d g f
  d   g f         d     d d     d d     f g     g
-------------------------------------------------
c     5 e   d   d         c
  e c e   d g g   g b 2
c     c c     g g f d 7 c   5 $
-------------------------------------------------

```

(4)

MUSIC BOARD

```
c   c d     c           f     e             c   c d
A 1   a 1 4 6           a     g a 1 3     e     g
      f     a           b   f e     g     A     g c
-------------------------------------------------
c f   g     f           c     c     a       f
$ a   1     a 1 1   4   6     4     b f a f a f a
e     d   e f           a     B   B f     a     c
-------------------------------------------------
g a 2 4     a     A   a     f     g   f
g A 2 4 2 4 3 f b   f   a 1 b     A a g f
g $   a     f       f       d     c d e f
-------------------------------------------------

```

(5)

MUSIC BOARD

```
c c c e a a a   g g g   f g g g   g g G a a a G
          5 5 5           4 4 4   3 3 3 4 4 4 4
c     g c e a   d       g f d d g     a f     e
-------------------------------------------------
g   b   b           g g f f e e e g g g g c
4   4   3
d   c   c       c   e g g c   g c   e g c c f c
-------------------------------------------------
  f f e d d d d d a a G G g     a a a d B B B

d   f g f   g   d   f g g f g f f   a B a   B
-------------------------------------------------
b b b b a           g g G G a a a g   b   b
                                  4   4   3
e   g b a g e c d   g g g d   g d   c   C
```

(6)

2.6 F BASIC 初步知识

先按 F5 键进入多功能菜单，再按数字键 1，即可进入 FBASIC 状态。首先进入如下“菜单”，读者可以通过选取[1]、[2]、[3]来“点菜”。选择[1]进入 FBASC；选择[2]进入 BG GRAPHIC 绘图程序；选[3]便退出 FBASIC。

```
F BASIC
1—BABIC
2—BG GRAPHIC
3—END
1、2、3KEY IN!!
```

▲进入 FBASIC

按数字 1 后进入 FBASIC。下面举一个程序的例子，读者并不需要搞懂它只要会玩就可以了。

〈程序 2.1〉

```
5  CLS ↓
10  SPRITE ON ↓
20  CGSET 1,0 ↓
30  FOR N=0TO 7 ↓
40  DEF MOVE(N)=SPRITE(0,N+1,3,255,0,0) ↓
50  NEXT ↓
60  MOVE 0,1,2,3,4,5,6,7 ↓
RUN ↓
```

下面解释一下这个例子。

5、10……60 是程序标号又称为行号，每个行号之后又有一些指令和参数。由指令和参数可以构成语句，上述程序中每一行只有 1 个语句。有时指令和语句并不能严格区分开来，比如：CLS、SPRITE ON 本身可叫指令，但在程序当中又作为语句使用。

每一行后面的 ↓，代表按回车键，如果不按回车键，程序不会换行。所以在每一行结束时必须按回车键。

RUN 表示运行程序，一键入：

```
RUN ↓
```

8 个玛丽沃向 8 方奔跑。

用[Pause]键可以令玛丽沃运动停下来。不过不会马上停下来，这里的原因后面将介绍。

读者可以键入下面语句替换程序中的语句。

```
20  CGSET 1,0
20  CGSET 1,2
```

每次用一条，后键入的语句将取代程序中同行号的语句。

改完后用回车键把光标下移到 60 行之后，键入：

```
LIST  ↓          ;看程序是否正确,
```

若正确，键入：

```
RUN  ↓           ;可以看出玛丽沃的颜色改变了。
```

读者也可以键入下面的语句看看有什么变化。

```
40  DEF MOVE(N)=SPRITE(1,N+1,3,255,0,0)↓
40  DEF MOVE(N)=SPRITE(2,N+1,3,255,0,0)↓
40  DEF MOVE(N)=SPRITE(3,N+1,3,255,0,0)↓
40  DEF MOVE(N)=SPRITE(4,N+1,3,255,0,0)↓
40  DEF MOVE(N)=SPRITE(5,N+1,3,255,0,0)↓
40  DEF MOVE(N)=SPRITE(6,N+1,3,255,0,0)↓
40  DEF MOVE(N)=SPRITE(7,N+1,3,255,0,0)↓
40  DEF MOVE(N)=SPRITE(8,N+1,3,255,0,0)↓
40  DEF MOVE(N)=SPRITE(N,N+1,3,255,0,0)↓
```

把 DEF SPRITE 的第一个参数分别用 1—8 和 N 来代替。

运行修改过的程序之后,发现玛丽沃,丽莎、苍蝇纷纷登场,我们把玛丽沃、丽莎等统称为卡通。各种卡通图案被印在封底。利用第 40 行语句都可将它们调出。这个程序体现了 FBASIC 的特点,应该仔细体会。后面将多次介绍这个程序。

▲程序修改方法

同计算板,打字板和音乐板。

▲在 FBASIC 中,功能键存入如下指令,可直接使用。

[F1] LOAD(M) [F2] PRINT [F3] GOTO

[F4] CHR $ ([F5] SPRITE [F6] CONT(M)

[F7] LIST(M) [F8] RUN(M)

▲从 FBASIC 退出

在没有行号的行中键入:

SYSTEM↓

便可以退回到菜单。注意,这时的英文必须是大写的。

从菜单中退出,只能用数字键[3],而不能用[ESC]键。

▲进入 BG GRAPHIC 绘图程序

在菜单中选数字[2],可以进入 BG 绘图程序。BG 绘图程序被用于绘制背景。在第 3.10 节有详细介绍这里从略。

▲从 BG GRAPHIC 绘图程序中退出

按[ESC]键

再按[Pause]键

便可以退回到菜单。

从菜单中退出,用数字[3]。

经过上述练习,读者可能已经积累了不少问题,下面的章节将一步步解决您的问题,并逐步把您带入 BASIC 天地中。

三、FBASIC 入门篇

3.1 独特的显示方式

FBASIC 是一种面向游戏的 BASIC 语言，显示方式便是其特点之一。FBASIC 中的显示画面由 4 层组成，如图 3.1 所示：

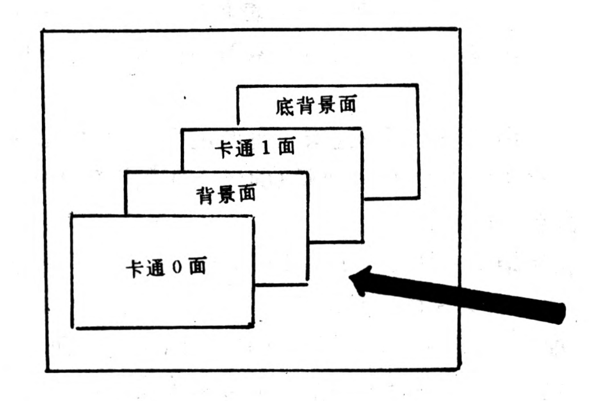


图 3.1

第一层称为卡通 0 面。用 SPRITE 指令可以把卡通图形调到这个面上来。SPRITE 指令需要有一套参数与之相配合使用，在后面将陆续介绍。卡通面的大小按点计算，其大小为：横（X 方向）是 256 点；纵（Y 方向）是 240 点。

第二层称为背景面。BASIC 指令和程序就被输入到这一层显示。背景图形，如高山、建筑物等也是建立在这一层上的。后面将要介绍的 BG 绘图程序就可把图形送入这一层中。背景面的大小按字符计算，横（X 方向）为 28 个字符，纵（Y 方向）为 24 行。

第三层称为卡通 1 面。在一些情况下，卡通图形需要隐藏在背景之后。这时就要用到 SPRITE 指令把卡通隐藏起来。这一层的大小同第一层。卡通 0 面和 1 面统称卡通面。

第四层称为底背景面。适合表示天空和海的效果。整个面用一种色表示，最初被设定成黑色。用指令可以改变底背景面的颜色。底背景面的大小为：横（X 方向）是 30 个字符；纵（Y 方向）是 30 行。

具备上述知识后有助于理解下面的例子。

3.2 卡通图案、背景图案、色彩代码

封底图 A 是卡通图案的示意。这些图案被预先存在学习卡中。在每个图案的四角有 4 个代码，这个代码与附录符号代码 A 中的代码完全对应。在需要某个卡通图案时，可以用特定的指令调用卡通图案的代码，即可调出对应的卡通图案。

为了表示出动画效果，每一个卡通又有许多姿势。比如，玛丽沃就有 7 个姿势，而乌龟则只有两个姿势。

为了表示出走步的效果至少应有两个姿势交替出现，即：

立正、走、立正、走······

可以看出乌龟只能完成“走”这一个动作。

一个卡通动作可以由两个或三个卡通姿势组成，也可以由一个卡通姿势组成。比如，“玛丽沃跳”就是由一个姿势组成动作。但是，这时的动作不会“动”。运行过第二章程序 2.1 的读者会发现，向四角运动的玛丽沃，都只能采用“玛丽沃跳”这一个姿势。换句话说，当只有一个姿势时，卡通动作

和姿势便没有区别。否则，两者是不同的概念。

卡通、卡通动作和卡通姿势是三个很重要的概念，三者在概念上有从属关系：

卡通姿势从属于卡通动作，卡通动作又从属于卡通。也就是说卡通一词的含义最广泛。

卡通姿势一词对应于卡通图案或卡通图形。

封 3 中又有两个图。一个是图形表 B，一个是色彩图表。下面分别介绍：

图形表 B 是用于绘制背景图案。左侧是图形表 B 的内容，一共有 13 组，一组又有 8 个。利用图形表 B 中的图形，可以搭出城墙、房子。图形表 B 右侧的机器人、山和星星、树和岛，也是用图形表 B 中的图形搭的。

根据不同的配色，图 B 中每一个图形还有三种颜色可以变化，即每一个图形共有 4 种配色。

每一个图形大小为一个字符，即 8×8 点阵（64 个点）。点阵的概念已经比较好懂，商店里的电子广告牌上的字，都是由点阵组成的。

关于图 B 的使用，读者请参见 3.10　BG GRAPHIC 绘图程序的使用一节。

色彩图表是为给卡通和背景着色所准备的。

FBASIC 为卡通面准备了三种不同色彩的屏幕面板，不同的面板用不同的代码，称为板代码，用 0，1，2 表示，用后面将要学习的 CGSET 指令，可以改变卡通面板的代码。当板代码改变时，所有卡通面上内容的颜色都将改变，下面的程序可以说明这一点。有 BASIC 经验的读者可以运行一下看看。没有 BASIC 经验的读者可以跳过而不影响学习。

程序〈3.1〉

```
5  CLS                                          ;清屏幕
10 SPRITE ON                                    ;打开卡通面
15 INPUT N                                      ;N 取 0,1,2
20 CGSET 1,N                                    ;设定板代码 0,1,2
30 DEF SPRITE N,(0,1,0,1,0)=CHR$(1)+CHR$(0)+CHR$(3)+CHR$(2)
                                                ;定义卡通姿势
40 SPRITE N,100+115*N,100                       ;把卡通在卡通面上显示出来，* 表示乘号
50 GOTO 15                                      ;跳回 15 行
```

运行上述程序可以发现，三个卡通（玛丽沃）的颜色被一起改变。

同理，FBSAIC 为背景面准备了两种不同色彩的屏幕面板，不同的面板用不同的代码，也称为板代码，用 0，1 表示。同样用 CGSET 指令指定。当板代码改变时，所有背景面上内容的颜色都将改变。

卡通面和背景面的板代码，都用 CGSET 指令指定。

无论是用于卡通面的板还是用于背景面的板，每一个板又分成 4 块，每一块也有一个代码称为配色代码。

卡通面的配色代码可以用 DEF SPRITE 指令和 DEF MOVE 指令指定。每一次只能根据指令的规定改变卡通面上特定卡通的配色。

背景面的配色代码用 COLOR 指令指定，COLOR 指令只能改变背景面中局部位置的配色。COLOR 指令一次只改变 16×16 点阵的配色。

配色代码指定的每一块又有二行三列数字，这些数字叫做颜色代码，用 PALET 指令改变。运用 PALET 指令之后，可以改变配色代码所代表的配色。比如，0 号配色代码的玛丽沃原来基本上

是红色的，用 PALET 指令后，可把 0 号配色代码的玛丽沃变为绿色的。

本节所介绍的内容在普通 BASIC 中没有，相对而言有一定难度，在读第三章中若有对代码不能理解的地方也可参考第四章有关指令的讲解。

3.3 让玛丽沃运动的方法

在 2.6 节程序 2.1 中，读者已经了解到玛丽沃可以八方奔跑。下面介绍另一种让玛丽沃运动的方法。

▲打开卡通面

键入 SPRITE ON　　　　；打开卡通面指令。

用这个指令可以打开卡通面。若卡通面被关闭，就看不见卡通了。

与打开卡通面指令相对应有关闭卡通面指令：

SPRITE OFF

用关闭卡通面指令，卡通面将关闭。不要键入 SPRITE OFF 指令。

▲定义玛丽沃的一种姿势

封底图 A 中玛丽沃共有 7 种姿势，这些姿势若要在程序中使用，一定要经过定义。定义卡通姿势的指令，还可以定义卡通姿势的反转姿势，图 3.2 便是"玛丽沃走 1"姿势的反转。定义卡通姿势的语句为：

键入 DEF SPRITE 0,(0,1,0,1,0)=CHR $ (1)+CHR $ (0)+CHR $ (3)+CHR $ (2)

图 3.2 表示编号和卡通图形的对应关系。CHR $ (　) 是字符串指令：

CHR $ (1)　对应左上

CHR $ (0)　对应右上

CHR $ (3)　对应左下

CHR $ (2)　对应右下

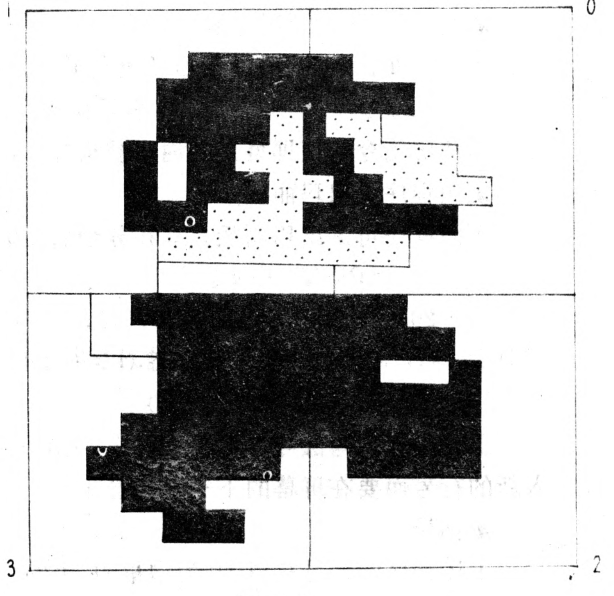


图 3.2

上述语句中等号的右边表示字符串"加"，即玛丽沃可由 4 串字符相加而成。字符串"加"不同于数值相加，比如，字符串(1)="ABC"，字符串(2)="DEF"两串相加的结果为：

ABCDEF

为了说明上的方便，把定义卡通姿势的上述语句改写成：

DEF SPRITE A,(B,C,D,E,F)=······

其中：A=0 表示定义的是 0 号卡通姿势，即姿势代码为 0。

B=0 表示 0 号姿势采用 0 号配色代码。

C=1 表示卡通图形由 16×16 点阵组成。

D=0 表示用卡通 0 面。

E=1 表示卡通按封底图 A 中给出的图形进行左右反转。试比较图 3.2 和封底图 A 中"玛丽沃走 1"。

F=0 表示卡通与图 A 给出的图形相同，不发生上下反转。

▲改变玛丽沃的位置

右图表示了屏幕画面的座标。(x,y)表示 x 方向和 y 方向的座标。下面指令可把玛丽沃移到座标(100,100)位置上。

键入 SPRITE 0,100,100　　　;0 是卡通姿势代码;100 是座标

若改变语句中的数据可以改变卡通图形的位置,如:

键入 SPRITE 0,150,150　　　;执行这条语句,卡通在比上条语句的位置偏右下的方向出现。

若要修改语句,可以利用方向键来移动光标到需要修改的地方,直接键入数字或符号按回车键即可。

▲直接语句方式与程序方式

上述几条语句(包括指令)都输入一条,执行一条,这种方式被称为直接语句方式。但是,对于一些情况,一条一条往计算机输入语句执行,未免太不方便,如果能把一件事:比如可以把 3.3 节所采用的语句,按顺序记忆在计算机中。做到一次输入而让计算机多次执行和连续执行。这种预先编好"节目",让计算机按顺序连续执行的方式,就是程序方式。

3.4 用程序让玛丽沃运动

在 2.6 节,曾经举过一个程序的例子。与 3.3 节比较可以发现,前者有标号后者无标号。这个标号给出了程序的顺序,可供计算机识别和执行之用。没有标号,计算机便不知道该执行那一句。那么什么是计算机程序呢?简单地说是按顺序排列起来的语句。下面把 3.3 节所用的例子用程序表示出来。在表示程序之前,先来做一下准备工作。

▲用 CLS 指令清理屏幕

经述上面输入指令,屏幕已经很乱了,这时可以执行清屏指令:

CLS ↓　　　;↓表示按回车键

用清屏指令可使屏幕上的内容被清掉。这时光标跑到左上角。

▲用程序调出玛丽沃

下面程序可调出玛丽沃。写在分号后面的是注释。

```
10  SPRITE ON ↓                        ;打开卡通面
20  DEF SPRITE 0,(0,1,0,1,0)=          ;定义卡通姿势代码为 0。
    CHR $ (1)+CHR $ (0)+CHR $ (3)+CHR $ (2) ↓
30  SPRITE 0,120,140 ↓                 ;120 是 x 座标,140 是 y 座标。
```

上述程序的写法与屏幕并不对应,在屏幕上,第 20 行语句要超过屏幕一行的长度,这时不要输入新的行号而要在屏幕的下一行接着输入。

▲标号

上述程序的标号是以 10 为增量的,目的是在语句中还可以插入语句。标号增量最小为 1,但不应该重叠。若重叠,后打入的语句将取代前面一条同标号的语句。标号也被称为行号。

▲用列表指令 LIST 列出程序清单

程序输入完并经过各种修改(修改方法同前)后,想看一看程序是否正确。可以用列表指令:

LIST ↓

执行这个指令,程序以"清单"的形式被列出。

▲用执行指令 RUN 执行程序

上述修改好的程序可以用执行指令

RUN↓

来执行，得出程序运行的结果。试与3.3节的结果比较一下，从中可以体会到程序方式的优点吧！

▲用清内存指令NEW清除程序

键入清内存指令

NEW↓

程序将被清除。用列表指令LIST再也列不出。所以在使用NEW指令前，一定要确认内存的所有内容是否都是无用的。

▲CLS和NEW指令的区别

CLS不清除内存的内容，只清屏幕。执行CLS之后，用LIST仍然可以列出存在内存的程序。而执行NEW则内存内容被清除，但屏幕不清除。

3.5 用变量改变玛丽沃的位置

▲变量

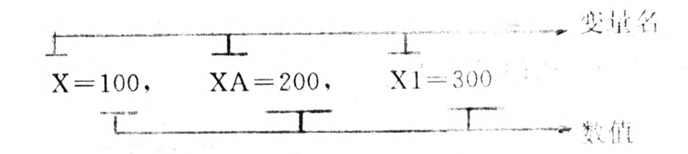
变量相当于可放置某种东西的盒子，不同的盒子有不同的代号以示区别。变量有不同的代号如，x，x1，xA，A$，B$等等。变量的代号称为变量名，而对于不同的BASIC语言，对变量名也有不同的规定。变量又分为数值变量和字符变量。分述如下：

(1) 数值变量

数值变量需以英文字母打头，长度可达255位，但是只识别前两位。如，ABC和ABDFE是相同的变量名。

变量的内容需用整数表示。下面举一个例子。

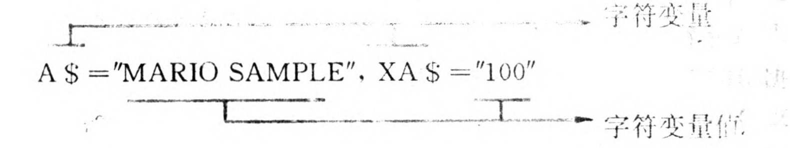
例3.1：



(2)字符变量

字符变量也需以英文字母打头，长度可达255位，但计算机只识别前两位。在字符之后加有“$”记号，用以和数值变量相区别。字符变量的内容要加上引号，请看下面的例子。

例3.2：



尽管100是数值，但在括号内作为字符变量的内容，被当作符号。

总之，字符变量是装符号的变量，数值变量是装数值的变量。字符变量不能比较大小，正像红色和兰色不能比较大小一样，但是字符变量可以相加。相加的结果不同于数字相加。

另外，不能用BASIC语言中的保留字，如NEW，RUN等作变量名。

在程序3.1中使用的变量CHR$(n)，在本书中归入字符变量类，并被称为字符函数。

▲给变量赋值

例 3.1 和例 3.2 就是给变量赋值的例子如：

A=100　　　　　　A $ ="MARIO SAMPLE"

上式的意义是，把等号右边的值赋给等号左边的变量。之所以采用赋值的说法，是因为等号两边在形式上可以不相等。如：

A=A+100

这个式子在 BASIC 中是合法的语句，而且这一语句还很有用处。在一定条件下可以起到累加器的作用。

对于没有被赋值的变量，计算机将自动赋给 0 值。

等式两边的类型不能错，比如下面的几个例子都是错误的，即在 BASIC 中被认为不合法。

例 3.3：

A="ABC"　　　　　　AI="123"

B $ =123　　　　　　BI $ =ABC

上述式子若输入到计算机，将自动地给出出错信息。

数值变量和字符变量取值范围都是有限制的。

数值变量取整数时的范围：(—32768～+32767)

字符变量取值的字符长度：(0～31)

这一点与通常的 BASIC 不同，希引起注意。

▲用变量改变玛丽沃的位置

程序 3.2 是利用变量让玛丽沃运动的程序。

〈程序 3.2〉

```
 5  CLS↓                                        ;清屏幕
10  SPRITE ON↓                                  ;打开卡通面
20  DEF SPRITE 0,(0,1,0,1,0)=                   ;组成 0 号卡通姿势，0 号卡通是玛丽沃
  CHR $ (1)+CHR $ (0)+CHR $ (3)+CHR
$ (2)↓
30  INPUT"Y=";Y↓                                ;输入 Y 方向的数值
40  INPUT"X=";X↓                                ;输入 X 方向的数值
50  SPRITE 0,X,Y↓                               ;0 号卡通在 X,Y 位置上出现
```

运行上述程序：

键入 RUN↓

屏幕出现：Y=?

键入数据：180↓

屏幕出现：X=?

键入数据：150↓

运行结果：玛丽沃出现在(150,180)座标的位置上。

不断运行上述程序，玛丽沃将不断运动。

▲输入指令 INPUT

利用 INPUT 指令可以把从键盘上键入的数字和符号送入变量中。基本格式为：

INPUT"字符",变量

引号中内容可以直接显示在屏幕上。紧跟在字符后出现的问号，是要求键入数据的标记。

3.6 提高玛丽沃运动效率的方法

▲用跳转指令 GOTO 移动玛丽沃

利用跳转指令 GOTO 可以不断地改变玛丽沃的位置。在程序 3.2 中加上一句并运行之。

60 GOTO 30↓ ；跳转到第 30 行

键入 RUN↓

屏幕出现：Y=? ；要求输入数据

键入数据：100↓

屏幕出现：X=? ；要求输入数据

键入数据：100↓

这时，玛丽沃在座标(100,100)位置上出现。同时，

屏幕出现：Y=? ；要求输入数据

键入数据：140↓

屏幕出现：X=? ；要求输入数据

键入数据：120↓

这时，玛丽沃出现在座标(120,140)位置上，同时

屏幕出现：Y=?

如果不断键入不同的数据，可使玛丽沃不断移动。与程序 3.2 相比，改变位置时不用每次执行 RUN 指令。

如果想停下来，可按一下 [Pause] 键。这时

屏幕出现：BREAK IN 30 ；表示在第 30 行被打断运行。

若还想继续执行程序，

键入 CONT↓

屏幕再次出现：Y=? ；要求输入数据

▲用循环指令 FOR～NEXT 让玛丽沃运动

给程序 3.2 加上两条语句：

40 FOR X=1 TO 200↓ ；

60 NEXT↓ ；第 40 行和第 60 行一起构成 X 变量的循环语句。

运行修改后的程序

键入 RUN↓

屏幕出现：Y=? ；要求输入数据

键入数据：100↓

运行结果：玛丽沃用同一姿势在第 100 行由左向右移动 200 次(200 个点)。

如果看不清楚运行结果，请插入语句：

55 STOP

运行后并输入 Y 值，再按 [F6]，可见玛丽沃一点一点前进但姿势不变。

修改第 40 行为：

40 FOR X=0 TO 200 STEP 2↓

STEP 2 表示 X 每一次变化 2 步即：0,2,4,6…

键入 RUN↓

屏幕显示：Y＝?

键入数据：80↓

运行结果：玛丽沃快速向右移动。

如果让Y方向也循环起来，则要修改第30行并加上第70行。

```
30  FOR Y=0 TO 200 STEP 5↓              ;STEP 5 表示一次变化 5 步，0，5，10…
70  NEXT↓
```

第30行和第70行一起构成Y循环语句。注意循环语句总是成对出现的。

键入RUN↓

屏幕出现：玛丽沃从左上角移动到右下角。

试把第40行的X和第30行的Y调换，看一下运行结果是什么？

修改后的程序见下面。

〈程序3.3〉

```
5   CLS
10  SPRITE ON
20  DEF SPRITE 0,(0,1,0,1,0)=CHR$(1)+CHR$(0)+CHR$(3)+CHR$(2)
30  FOR Y=0 TO 200 STEP 5
40  FOR X=0 TO 200
50  SPRITE 0,X,Y
60  NEXT
70  NEXT
```

▲多重循环

循环语句是让变量X(Y)循环一次增加一次值，增加的多少取于步长STEPn的大小，n越大，一次循环X(Y)增加的值就越多。当X(Y)超过其终值，如上列中的200，循环就结束。而循环变量不一定局限于X和Y，用其它字母也可以。

程序3.3中由X、Y构成两重循环。X在内，Y在外。当X循环200次，Y才循环一次。也就是说，多重循环的次序是先内后外。另外，FOR与NEXT总是成对出现，当程序执行到NEXT指令就回去找FOR语句，当循环变量的值大于所设定的值，比如200，就从这一循环中转出。关于循环语句可参见第四章。

3.7 用操纵器控制玛丽沃运动

▲利用程序调出玛丽沃各种姿势

在封底上玛丽沃有7种姿势，用下面程序可以调出3种并得出3个左右对应的反转图形。

〈程序3.4〉

```
5   CLS↓
10  SPRITE ON↓
15  CGSET 1,0↓
20  DEF SPRITE 0,(0,1,0,1,0)=CHR$(1)+CHR$(0)+CHR(3)+CHR$(2)↓
21  DEF SPRITE 1,(0,1,0,0,0)=CHR$(0)+CHR$(1)+CHR$(2)+CHR$(3)↓
22  DEF SPRITE 2,(0,1,0,1,0)=CHR$(5)+CHR$(4)+CHR$(7)+CHR$(6)↓
23  DEF SPRITE 3,(0,1,0,0,0)=CHR$(4)+CHR$(5)+CHR$(6)+CHR$(7)↓
24  DEF SPRITE 4,(0,1,0,1,0)=CHR$(21)+CHR$(20)+CHR$(23)+CHR$(22)↓
```

```
25  DEF SPRITE 5,(0,1,0,0,0)=CHR$(20)+CHR$(21)+CHR$(22)+CHR$(23)↓
30  SPRITE 0,100,100↓
40  SPRITE 1,150,100↓
50  SPRITE 2,100,150↓
60  SPRITE 3,150,150↓
70  SPRITE 4,100,50↓
80  SPRITE 5,150,50↓
```

键入 RUN↓

可以得到玛丽沃的 6 个不同姿势，试与封底比较。

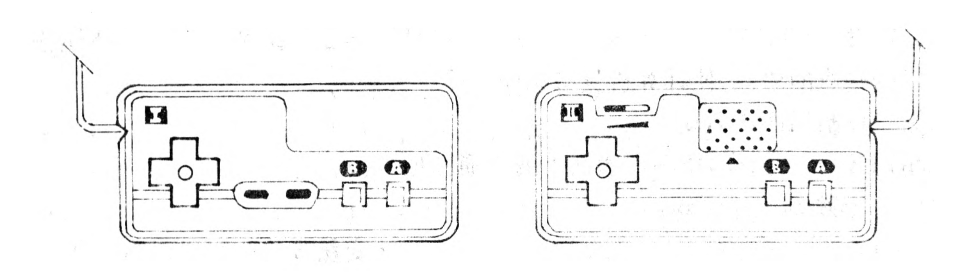


图 3.3

▲**操纵器指令**

操纵器如图 3.3 所示。上面有十字形的方向键，1 号操纵器上有选择键(SELECT)和起动键(START)。在 1 号和 2 号操纵器上分别有按钮 A 和 B。在 2 号操纵器上还有麦克风和麦克风音量控制。下面分别介绍控制方向键指令 STICK 和控制其它键的指令 STRIG。

(1)STICK 指令

STICK 指令是 FBASIC 的专用指令。

1 号操纵器对应于:STICK(0)

2 号操纵器对应于:STICK(1)

用 STICK 指令可以测试操纵器十字形方向键按下与否。十字形方向键的每一个方向被按下都产生一个数值。STICK 指令可以接收这个数值。十字形方向键如图 3.4 所示。



图 3.4

令 S=STICK(0)或 S=STICK(1)

若按下十字键右侧，S=1；

若按下十字键左侧，S=2；

若按下十字键下侧，S=4；

若按下十字键上侧，S=8；

利用 STICK 指令和十字键的配合，可使卡通图形上下左右移动。

(2)STRIG 指令

STRIG 指令是 FBASIC 专用指令。

1 号操纵器对应于:STRIG(0)

2 号操纵器对应于:STRIG(1)

利用 STRIG 指令可把操纵器的 4 种按键:**起动、选择、A 和 B**,以不同的数值传给主机。令 T=STRIG(0)或 T=STRIG(1)若:

若起动键被按下:T=1

若选择键被按下:T=2

若 B 钮被按下:T=4

若 A 钮被按下:T=8

STRIG 指令主要是控制主机起动,选择游戏项目,控制 A 钮和 B 钮在游戏中的动作等等。

▲条件指令 IF~THEN

条件指令的意思是,如果…则…,请看例句:

```
40  IF S>2 THEN 100              ;如果 S>2 则跳到第 100 行;否则顺序执行。
50  IF S=2 THEN X=X-1            ;如果 S=2 则让 X=X-1;否则顺序执行。
```

由条件指令构成的语句,使计算机具有判断功能。

▲用操纵器控制玛丽沃运动

选用 SPRITE OFF 指令,使在卡通面上的卡通图形**消失**。

利用程序 3.4,增加下述语句:

```
26   X=100↓                      ;让变量 S 为 1 号操纵器变量。
30   S=STICK(0)↓                 ;若 S>2 跳转到 100,说明上下方向移动无
40   IF S>2 THEN 100↓             效。
50   IF S=2 THEN X=X-1↓          ;若 S=2 则让玛丽沃向左一步。
60   IF S=1 THEN X=X+1↓          ;若 S=1 则让玛丽沃向右一步。
70   IF X>250 THEN X=X-240       ;若 X>250 则让玛丽沃回到左边起始位
80   IF X<5 THEN X=X+245        置。
90   SPRITE 0,X,150↓             ;若 X<5 则让玛丽沃 从右边出现。
100 GOTO 30                      ;只让 0 号卡通(玛丽沃一种姿势)运动。
键入 RUN↓                        ;跳转到 30,等待接受按键。
;让变量 X=100。
```

玛丽沃的移动将受 1 号操纵器控制。

第 21—第 25 行的程序,在本程序中并未实际使用,删去后仍不影响程序的运行。

▲让卡通图形消失的另一种方法

利用 SPRITE OFF 可使卡通消失。如果只让一个或几个卡通消失,SPRITE OFF 指令将无能为力。这时可用:

```
6  SPRITE1:SPRITE2:SPRITE3:SPRITE4:SPRITE5
```

把这条语句插入程序 3.4,并加上上述从 26—100 行程序,即使不用 SPRITE OFF 指令也可让 1~5 号卡通消失。

SPRITEn 实际上是让 n 号卡通消失的指令。

▲冒号的作用

在上述第 6 行语句中采用了冒号，采用冒号可以在同一行号下打入多条语句。但是，一个行号下(包括冒号在内)，所能键入的字符数不能超过 256 个。

▲用操纵器控制玛丽沃动作

下面的程序可以使玛丽沃左右行走。在输入程序之前，用 LIST 指令进行列表，看内存中还有没有程序，若有程序需要删除，可用 NEW 指令。要清除画面的话，可以用 CLS 指令。程序 3.5 所表示的是让玛丽沃变换走步姿势左右行走的程序。注释写在程序的右边。由于程序较大，这里分析一下程序的执行过程。

当键入 RUN 之后，玛丽沃用 1 号动作出现在座标(120,140)位置上，这便是第 50 行语句的功能。当无键按下时，玛丽沃将会停在那里不动。因为根据顺序执行原则，计算机执行完第 50 行后便执行第 60 行。在第 60 行有两条语句，一条是赋值语句 S=STICK(0)，另一条是条件语句 IF S=0 THEN 60。当无键按下时 S=0，这时根据条件语句：如果 S=0 则跳回 60 行。所以无键按下时，程序将在第 60 行循环执行第 60 行的语句。

〈程序 3.5〉

```
5 CLS                                          ;清屏幕
10 SPRITE ON                                   ;打开卡通面
20 DEF SPRITE0,(0,1,0,1,0)=CHR$(1)+CHR$(0)+CHR$(3)+CHR$(2)
21 DEF SPRITE1,(0,1,0,0,0)=CHR$(0)+CHR$(1)+CHR$(2)+CHR$(3)
22 DEF SPRITE2,(0,1,0,1,0)=CHR$(5)+CHR$(4)+CHR$(7)+CHR$(6)
23 DEF SPRITE3,(0,1,0,0,0)=CHR$(4)+CHR$(5)+CHR$(6)+CHR$(7)
                                               ;20—23 表示四种卡通姿势(玛丽沃)
30 READ X,Y,A,B,C,D,                           ;30—40 是读数据语句
40 DATA 120,140,1,3,0,2                        两者一一对应需配对出现
50 SPRITE A,X,Y                                ;1 号姿势出现在(120,140)位置上
60 S=STICK(0):IF S=0 THEN 60                   ;设定 1 号操纵器变量 S,S=0 则跳回 60 行
70 IF S>2 THEN 60                              ;限制上、下方向运动
80 X=X+2:IF S=2 THEN X=X-4                     ;确定走步步距，一次 2 个点
90 IF S=1 THEN 1000                            ;向右走则跳到 1000
100 IF S=2 THEN 2000                           ;向左走则跳到 2000
1000 PAUSE 5:SPRITE C,X,Y:SWAP C,D             ;2000—2020 向右走子程序
1010 SPRITE A:SPRITE B:SPRITE C
1020 GOTO 60
2000 PAUSE 5:SPRITE A,X,Y:SWAP A,B             ;2000—2020 向左走子程序
2010 SPRITE A:SPRITE C:SPRITE D
2020 GOTO 60
```

第 70 行的作用是限制玛丽沃上下运动。我们知道上下运动时的 S 值将大于 2，而如果 S>2 则程序跳回 60 行，这样就限制了玛丽沃上下运动。

控制向右走的语句有 80，90 及 1000 到 1020 行。下面分析向右走的过程。在第 80 行有：

X=X+2

X 的初始值是 120，即玛丽沃不动的 X 方向的座标，此时 80 行为 X=120+2，可向右移动两个点。向右移动 S=1(十字按键右支被按下)，从而执行第 90 行语句，第 90 行是条件跳转语句，若 S=1 则跳转到 1000 行。第 1000 行到 1020 是控制玛丽沃右行的程序。

第1000行由三个语句构成下面分别介绍。

PAUSE 5 是一个暂停语句，表示暂停5个单位时间。

SPRITE C,X,Y 表示显示0号姿势在座标(122,140)位置上。

SWAP C,D 是一个交换语句，表示C和D的数值互相交换。

C和D是在30和40行语句被赋值：C=0,D=2。

第1010行是让无关的玛丽沃姿势消去。A和B是控制向左运动的，当然应该消去。消去C实际上是消去D的值。C和D代表了两个玛丽沃姿势，当持续按十字键右支，C和D将交替出现，为了表示走步效果，只能去D值。C和D代表了两个玛丽沃姿势，当持续按十字键右支，C和D将交替出现，为了表示走步效果，只能留一个玛丽沃在屏幕上，要将走过的玛丽沃姿势消去，交换语句SWAP C,D和SPRITE C的使用便可完成这一要求。

第1020是跳转语句，跳转到第60行。

向左行走的分析方法同上，有关的语句有80、100、2000到2020行。读者可以自行分析一下。

▲读数据指令 READ～DATA

在READ语句有变量，在DATA语句有数据。变量与数据一一对应，并可放在程序的任何地方。程序3.5中第30行至40行的读数据语句可以等效如下语句：

```
30 X=120:Y=140:A=1:B=3:C=0:D=2
```

▲用操纵器控制玛丽沃器四方行走

当理解了程序3.5后下面的程序便不难理解。上下方向活动的指令，在100行、130行和3000到3020行中处理。101行到104行是控制玛丽沃位置的语句。

〈程序3.6〉

```
5   CLS                                        ;清屏幕
10  SPRITE ON                                  ;打开卡通面
15  CGSET 1,0                                  ;确定玛丽沃的颜色
20  DEF SPRITE 0,(0,1,0,1,0)=CHR$(1)+CHR$(0)+CHR$(3)+CHR$(2)
21  DEF SPRITE 1,(0,1,0,0,0)=CHR$(0)+CHR$(1)+CHR$(2)+CHR$(3)
22  DEF SPRITE 2,(0,1,0,1,0)=CHR$(5)+CHR$(4)+CHR$(7)+CHR$(6)
23  DEF SPRITE 3,(0,1,0,0,0)=CHR$(4)+CHR$(5)+CHR$(6)+CHR$(7)
24  DEF SPRITE 4,(0,1,0,0,0)=CHR$(20)+CHR$(21)+CHR$(22)+CHR$(23)
25  DEF SPRITE 5,(0,1,0,1,0)=CHR$(21)+CHR$(20)+CHR$(23)+CHR$(22)
26  X=100                                      ;20—25构成玛丽沃6种姿势
30  READ X,Y,A,B,C,D,E,F                       ;30—40为读数据
40  DATA 120,140,1,3,0,2,4,5
50  SPRITE A,X,Y
60  S=STICK(0):IF STRIG(0)<>0 THEN             ;若起动键、选择键、A钮和
    END
65  IF S=0 THEN 60                             ;B钮任意一个被按下，程序结束。
70  IF S>2 THEN 100
80  X=X+2:IF S=2 THEN X=X-4                    ;左右行走
90  GOTO 101
100  Y=Y+2:IF S=8 THEN Y=Y-4                   ;上下行走
```

```
101  IF X>255 THEN X=X-252              ;让玛丽沃从左边出来
102  IF X<3 THEN X=X+252                ;让玛丽沃从右边出来
103  IF Y>240 THEN Y=Y-237              ;让玛丽沃从上面出来
104  IF Y<3 THEN Y=Y+237                ;让玛丽沃 从下面出来
110  IF S=1 THEN 1000
120  IF S=2 THEN 2000
130  IF S>2 THEN 3000
1000  PAUSE 5:SPRITE C,X,Y:SWAP C,D
1010  SPRITE A:SPRITE B:SPRITE C:SPRITE E:SPRITE F
1020  GOTO 60                      ;1000—1020 向右行走程序
2000  PAUSE 5:SPRITE A,X,Y:SWAP A,B
2010  SPRITE A:SPRITE C:SPRITE D:SPRITE E:SPRITE F
2020  GOTO 60                      ;2000—2020 向左行走程序
3000  PAUSE 5:SPRITE E,X,Y:SWAP E,F
3010  SPRITE A:SPRITE B:SPRITE C:SPRITE D:SPRITE E
3020  GOTO 60                      ;3000—3020 向上下行走,但是玛丽沃只能头朝上行
走。
```

3.8 用 MOVE 系列指令控制玛丽沃八方行走

在 2.6 节程序 2.1 中已经见到玛丽沃八方行走的例子。这里先介绍其中用过的两条指令然后分析该程序的工作过程。

▲用 DEF MOVE 指令定义卡通动作及运动方式

DEF MOVE 是定义卡通动作及运动方式的指令,其结构如下:

DEF MOVE(n)=SPRITE(A,B,C,D,E,F)

各参数含义为:

A:表示卡通代码 0—15,0 号代表玛丽沃,1 号代表丽莎,2 号代表苍蝇……

B:表示卡通运动方向(0—8)其中 0 代表静止,如图 3.5 所示。

C:卡通运动速度:

1 代表最高,即每秒移动 60 个点

255 代表最低,即 255 秒移动 60 个点

D:表示卡通运动的范围(1—255)点

E:表示卡通出现在卡通 0 面还是卡通 1 面(0—1)

F:表示卡通的配色(0—3)

n:卡通动作代码(0—7),不同的卡通其动作有多有少。如玛丽沃动作较多,而乌龟的动作较少。

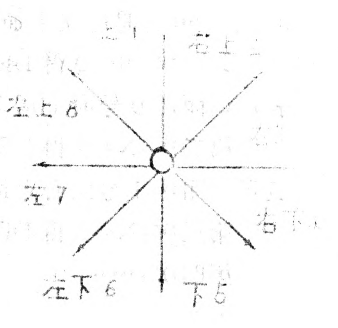


图 3.5

DEF MOVE(n)是个功能很强的指令,它可以定义卡通的动作。前面说过,一个动作可以由几个姿势组成,用 DEF SPRITE 只能定义卡通姿势,要让卡通有走步动作,要用一段程序,而 DEF MOVE(n)所定的就已经是动作了,根据不同的运动方向,DEF MOVE(n)自动给出应有的动作,但是,对于姿势较少的卡通如乌龟,无论在那一个方向上都只能有“走”这一个动作。

▲MOVE 指令可使卡通开始运动

MOVE 指令可使 DEF MOVE 指令所定义的卡通开始运动。如：

MOVE 0,1,2,3,4,5,6,7

这条指令可使 8 个方向的卡通同时开始运动。运动的范围被在 DEF MOVE 指令 D 项中设定。MOVE 指令中的代码(0—7)可以任选设定,如：

MOVE 1　　　　;让 1 号卡通运动

MOVE 0,3,6　　　　;让 0,3,6 号卡通运动

注意:MOVE 指令须与 DEF MOVE 配合使用,只有当 DEF MOVE 定义了 8 个方向的卡通,用 MOVE 指令才能从 8 个方向中任选一个或几个。读者可以改变 MOVE 指令中参数和 DEF MOVE 中 B 项参数比较一下结果。

▲玛丽沃八方行走程序分析

把程序 2.1 重写如下：

首先键入 NEW 再键入程序 3.7

〈程序 3.7〉

```
 5  CLS↓                                   ;清屏幕
10  SPRITE  ON↓                            ;打开卡通面
20  CGSET 1,0↓                             ;确定卡通颜色
30  FOR N=0 TO 7↓                          ;定义 8 个方向卡通动作。
40  DEF MOVE (N)=SPRITE (0,N+              ;方向参数 B 是可变的
1,3,
    255,0,0)↓
50  NEXT↓                                  ;让 8 个 DEF MOVE(n)同时起动
60  MOVE 0,1,2,3,4,5,6,7↓                  ;反复执行第 60 行
70  GOTO 60↓
```

下面分析程序 3.7

30 行—50 行是定义卡通 8 个动作,程序在 30 行—50 行要循环 8 次(0—7)。

第一次:N=0 得 DEF MOVE(0)=SPRITE(0,1,3,255,0,0)

表示玛丽沃 0 号向上运动。这里对应 MOVE 0。

第二次:N=1 得 DEF MOVE(1)=SPRITE(0,2,3,255,0,0)

表示玛丽沃 1 号向右上运动。这里对应 MOVE 1

第三次:N=2 得 DEF MOVE (2)=SPRITE (0,3,3,255,0,0)

第四次:……

·

·

·

第八次:N=7 得 DEF MOVE(7)=SPRITE (0,8,3,255,0,0)

通过上述 8 次循环可以定义 8 个玛丽沃动作。等式右边的含义是：

以 1 号为例：

……(1)=SPRITE (0,2,3,255,0,0)

表示:1 号,玛丽沃,向左上方,以 3 单位速度,可以移动到 255 点,并在卡通 0 面上出现,玛丽沃采用 0 号配色代码。8 次循环之后程序进入第 60 行。

第60行的MOVE指令有0—7个参数分别对应于DEF MOVE (0)、DEF MOVE(1)、…，若只让DEF MOVE(0)所定义的卡通运动，用MOVE 0即可控制。若让DEF MOVE(1)所定义的卡通运动，用MOVE 1控制。若让8个定义的卡通都动作，只能用第60行的语句了。MOVE指令实际上是控制DEF MOVE(n)指令的“开关”，由MOVE指令控制让哪一个DEF MOVE(n)有效。同样，若是没有让DEF MOVE(n)定义任何内容，就是让“开关”打开也得不到任何东西。

第70行的作用是反复执行第60行，即反复起动MOVE指令，因为若不反复起动当DEF MOVE(n)所定义的动作做完后，卡通便停下不动了。

要想打断上述程序请用 Pause 键。

理解了上述程序下面几个程序便不难理解。

▲8种不同卡通向8个方向运动

8种不同的卡通是玛丽沃、丽沙、苍蝇、飞鱼、企鹅、火球、车、太空站。

在程序3.7中改写第40行为：

```
40 DEF MOVE (N)=SPRITE(N,N+1,3,255,0,0)↓
```

与程序3.7第40行比较，只是把SPRITE中的第一个参数变成了N，当循环8次后，可得

```
…=SPRITE(0……)
…=SPRITE(1……)
…=SPRITE(2……)
  ……
…=SPRITE(7……)
```

从而使0—7个卡通可以出场。

键入RUN↓，可以看到玛丽沃、丽莎、苍蝇等向八方运动。

▲让八个玛丽沃直线前进

在程序3.7中加入下面程序

```
40  DEF MOVE(N)=SPRITE(0,3,N+1,255,0,0)↓
55  FOR N=0 TO 3:POSITION 2* N+1,120,180:NEXT↓
70  END
```

;用END指令表示程序结束，程序运行到这一行将终止。

键入RUN↓，可以看见8个玛丽沃一起向右走。

▲位置指令POSITION的作用

POSITION指令，决定开始运动的初始座标。指令格式是：

POSITION　n,X,Y

n:卡通姿势代码(0～7)

X:水平方向座标(0～255)

Y:垂直方向座标(0～255)

理解了POSITION位置指令，下面分析上述第40行、55行和70行的作用。

第40行中的变量是速度，8个卡通动作的速度为1～8,0号动作的速度最快，8号动作最慢。其它参数固定，从而运动方向被固定在右3方向。

第55行确定了4个动作1,3,5,7在座标(120,180)位置上开始起动。另4个动作在中线上开始起动。式2*N中"*"，表示乘号。

第70行是一个结束行。

▲让玛丽沃停止

在让玛丽沃直线运动基础上再加上以下几句，可使玛丽沃停止。

```
70  PAUSE↓                          ;暂停，可按任意一键。
80  CUT 0,1,2,3,4,5,6,7             ;使 0—7 号玛丽沃停止
90  END
键入 RUN↓
```

8 个玛丽沃开始行走，当按任意键后 8 个玛丽沃停止。

若输入：

```
MOVE 0,2↓                    ;在 CUT 指令之后指行 MOVE 指令
```

0 号和 2 号玛丽沃开始运动。

若改写第 80 行，键入：

```
80  CUT 0,3↓
    键入 RUN↓
```

8 个玛丽沃开始运动。当按空格键后，0 号和 3 号被停止，而其它玛丽沃照常行走。

▲让玛丽沃消失

若在上述程序中采用下面的语句可以让玛丽沃消失。

```
80  ERA 0,1,2,3,4,5,6,7↓            ;改写第 80 行，让 8 个玛丽沃消失。
键入 RUN↓
```

8 个玛丽沃开始行走，按空格键后 8 个玛丽沃都消失了。

若输入

```
MOVE 0,1↓              ;在 ERA2 指令之后执行 MOVE 指令
```

0 号和 1 号玛丽沃开始行走。

若改写第 80 行键入：

```
80  ERA 3,4↓
键入 RUN↓
```

8 个玛丽沃开始行走。当按空格键后，3 号和 4 号玛丽沃消失了。

▲让玛丽沃跳

下面这个程序可以让玛丽沃跳。

〈程序 3.8〉

```
  5  CLS↓
 10  SPRITE ON↓
 20  CGSET 1,0↓
 30  FOR N=2 TO 4↓
 40  DEF MOVE (N)=SPRITE(0,N,1,20,0,2)↓
 50  NEXT↓
 60  MOVE 3↓
 70  IF MOVE (3)=-1 THEN 70↓
 80  ERA3:POSITION 2,XPOS (3),YPOS (3):MOVE 2↓
 90  IF MOVE (2)=-1 THEN 90↓
100  ERA 2:POSITION 4,XPOS (2),YPOS (2):MOVE 4↓
```

```
110  IF MOVE (4)=-1 THEN 110
120  ERA 4:POSITION 3,XPOS (4),YPOS (4)↓
130  GOTO 60↓
```

在分析上述程序之前，先介绍 MOVE(n)、XPOS 和 YPOS 的作用。

▲MOVE(n)指令用于运动状态测试

基本格式：MOVE (n)=-1 或 0

n：表示在 DEF MOVE(n)中定义的卡通动作代号。

-1：表示在 DEF MOVE(n)中所定义的卡通仍在运动。

0：表示在 DEF MOVE(n)中所定义的卡通已经运动到终止位置。

以程序 3.8 为例介绍 MOVE(n)的作用。

在第 40 行中指定 3 号玛丽沃运动范围值是：20

在第 60 行指定 3 号玛丽沃运动。

在第 70 行指定 3 号玛丽沃只要没有运动到 20，MOVE(3)就将等于-1，程序在 70 行反复循环；当 MOVE3 运动到 20 时，MOVE(3)=0，程序进入到 80 行。

通过上述分析可知 MOVE(n)是一条很有用的测试指令。

▲XPOS 和 YPOS 是自动定位指令

在 DEF MOVE(n)中定义的卡通其运动位置的确定可用：

XPOS(n)确定 n 号卡通动作的 X 方向(水平方向)的座标。

YPOS(n)确定 n 号卡通动作的 Y 方向的座标。

以程序 3.8 为例介绍 XPOS 和 YPOS 的作用。

当 MOVE (3)=0 时程序进入到第 80 行：

```
80  ERA 3:POSITION 2,XPOS (3),YPOS(3):MOVE 2
```

首先让 3 号玛丽沃消失，让 2 号玛丽沃的起始位置为 3 号玛丽沃消失的位置，让 2 号玛丽沃运动。

理解了 60 行至 80 行程序，程序 3.8 便不难理解。90 行至 100 行和 110 行至 120 行的作用原理同 60 行至 80 行。程序的执行过程可以用图 3.6 表示

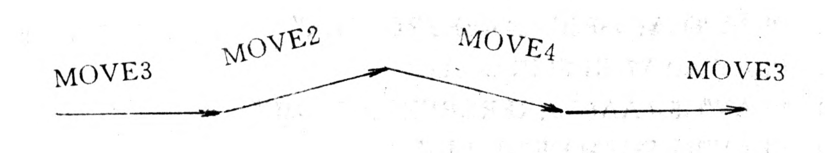


图 3.6　整个跳跃的过程

给程序 3.8 可以加上音乐效果，如

```
65  PLAY "T104C1B1DEG1CDE1"
85  PLAY"03CDE1G1A"
键入 RUN↓
```

将会使玛丽沃在屏幕中间跳起，着地，行走，同时伴有音乐。

▲用 MOVE 系列指令让玛丽沃四方移动

程序 3.9 是让玛丽沃四方运动的程序，该程序中没有新的指令，请读者自行分析。

〈程序 3.9〉

```
  5  CLS
 10  SPRITE ON
 20  CGSET 1,0
 30  FOR N=0 TO 7
 40  DEF MOVE (N)=SPRITE (0,N,1,3,0,0)
 50  NEXT
 60  S=STICK (0)
 70  IF S=0 THEN N=0:GOTO 120
 80  IF S=1 THEN N=3:GOTO 120
 90  IF S=2 THEN N=7:GOTO 120
100  IF S=4 THEN N=5:GOTO 120
110  IF S=8 THEN N=1
120  IF MOVE (M)=-1 THEN 120          ;开始 M=0,即不赋值变量为 0。
130  IF M=N THEN 160
140  ERA M:POSITION N,XPOS (M),YPOS (M):M=N
150  MOVE N:GOTO 60
160  MOVE M:GOTO 60
```

3.9 计算机演奏的例子

先用 NEW 指令清除内存。键入程序 3.10。音乐的名字叫做:"美国羊"。

〈程序 3.10〉

```
 5  CLS
 6  LOCATE 8.12:PRINT"AMERICAN SHEEP"
10  PLAY"M1Y2V7T3:M1Y1V5T3:M1T3"
20  PLAY"O2A6G3F5G:O2R3FCEDCEC:O1F7C"
30  PLAY"A5AA7:RFCFRCO1AO2C:FC"
40  PLAY"G5GG7:RECERCO1GO2C:O2CO1G"
50  PLAY"A5O3CC7:RF5AO3C3AG:FC"
60  PLAY"O2A6G3F5G:O2RFRERDRC:FC"
70  PLAY"AAA7:RFFFRCCC:FC"
80  PLAY"G5#AA6G3:RERGRERC:O2CO1G"
90  PLAY"F9:FCFAO3F7:F5CF7"
```

键入 RUN

开始自动演奏。

3.10 BG GRAPHIC 绘图程序的使用

BG GRAPHIC 是一种绘图程序，用于绘制背景。BG 绘图程序采用菜单结构，可以通过方向键 [↑][↓]……等来挑选 BG 程序所提供的功能，即"点菜"。

下面开始介绍。

▲进入 BG 程序的方法

首先需确认计算机是否在 FBASIC 程序设计状态，在 FBASIC 状态，请键入：

SYSTEM

这时出现菜单：

（见下页）

需要选取 1,2,3 来点菜。选 1 是进入 BASIC，选 2 是进入 BG 程序。选 3 便是结束。

▲屏幕上各种参数的含义

```
          F   BASIC

 1——BASIC
 2——BG GRAPHIC
 3——END

 1,2,3   KEY   IN   !!
```

当按下数字键 2 时，计算机进入 BG 程序状态，在屏幕左上角出现一个方块：

■

这个方块是光标。

在屏幕的下面出现一排符号：

X:00 MODE 0…………（——设计图形所使用的一组图形符号）

Y:00 SELECT ^ （——图形符号选择箭标）

下面分别介绍：

（1） 座标 X Y

光标在屏幕上移动，每个位置均有座标，X 表示横座标，Y 表示纵座标。当光标移动时 XY 值随之变化。

X 变化的范围 00～27，即 28 个字符；

Y 变化的范围 00～20，即 21 个字符。

可以看出，BG 程序同 BASIC 程序均采用字符座标，而且 BG 程序和 BASIC 程序均利用背景面，卡通面与背景面座标之间的关系如下：

x=(X * 8)+16

y=(Y * 8)+24

其中 x y 是卡通面的座标，X Y 是背景面的座标，* 表示乘号。在上述范围内，卡通可以重叠在背景上。

（2） MODE 0

MODE 是符号配色的指令，配色代码有 4 种(0～3)，配色表可参考封 3 的图表。配色可在 16×16

点阵内进行，16×16 点阵即 4 个字符的大小。

用 RETURN 键可以改变图形的颜色。

（3） 设计图形所使用的符号

设计图形的符号一共有 104 种，在封 3 图 B 中表示。一般一次显示出一组 8 个，如还要选出其它图形，用 PgDn 键。用 PgDn 键可从上到下选择 13 组图形，用 PgUp 键可以从下到上选择 13 组图形。

(4) 图形选择箭标

箭标∧被指在某一符号之下，按[DEL]键，箭标∧向右移动；按[INS]键，箭标∧向左移动。箭标所指的符号，是可以被输入到屏幕上的。

▲用于操作的指令

当按下[ESC]键在屏幕的左上角出现菜单：

SELECT
COPY
MOVE
CLEAR
FILE
CHAR

利用[↑][↓]键来移次箭标^来选择上述指令。当按下空格键后，表示选择完毕进入设计背景。

下面分别介绍各指令的作用。

(1)SELECT(选择)

SELECT 具有选择图形符号的作用，在 SELECT 状态下还可以完成配色等一系列操作。操作过程如下。

- 按[PgUp]键，或按[PgDn]键，选择一组 8 个图形。
- 按[DEL]键和[INS]键移动箭标选择所需图形。
- 按回车键[RETURN]改变要显示符号的颜色(配色共有 4 种)。
- 按方向键[↑][↓][→][←]键将光标■移到要显示图形的位置上。
- 按空格键，使所选定的符号显示在光标位置上。这时光标会自动后移，若要连续输入所选定的符号，可连续按空格键。
- 用符号[D]键，可以清除在光标位置的符号。

(2) COPY (复制)

COPY 指令可把已经选出的符号复制到其它指定位置。操作过程如下：

- 用[ESC]键进入指令菜单。
- 用[↓]移动箭标∧到 COPY。
- 按空格键进入到 COPY 状态。
- 按[↑][↓][→][←]键使光标移到母位(要把母位上的图形复制到它处)。
- 若按[INS]键，符号便可任意移动。
- 按[↑][↓][→][←]，光标移动到子位(欲接受图形的位置)。
- 位置决定后，按[DEL]键，图形便显示在光标位置上。同样地，用[↑][↓][→][←]和[DEL]键，可以连续地复制图形符号。

(3) MOVE(移动)

MOVE 指令具有把已存的符号移动到别处的功能。操作过程如下：

- 按[ESC]键显示指令菜单。
- 按[↓]键入选择 MOVE 指令。
- 按空格键进入到 MOVE 状态。

· 按[↑][↓][→][←]键将光标移动到欲被移动的符号上。

· 若按一次[INS]键,符号成可自由移动状态。若按两次[INS]键,符号便消失。

· 按[↑][↓][→][←]键,符号可移动到预定位置。

· 位置决定后按[DEL]键,符号可显示在光标位置上。

(4) CLEAR(清除)

CLEAR 指令具有消除屏幕上所有图形符号的功能。操作过程如下:

· 按[ESC]键显示指令菜单

· 按[↓]键选择 CLEAR 指令

· 按空格键,屏幕上的内容被清除。

· 清除后自动返回到选择状态。

(5) FILE(文件)

FILE 指令用于保存和取出磁带上的文件。操作过程参见 3.12 节。

(6) CHAR(符号)

CHAR 指令具有显示键盘上英文字母、符号和数字功能。操作过程如下:

· 按[ESC]进入指令菜单。

· 按[↓]键以选择 CHAR 指令

· 按空格键进入 CHAR 指令状态,可直接从键盘上把数字、英文字母、符号等输入到光标位置。

3.11 BASIC 程序与背景的连接

从 3.1 节至 3.10 节,通过举例已经初步了解了 FBASIC 程序控制卡通运动的方法,也了解了用 BG 绘图程序绘制背景图形的方法。当把卡通和背景连接在一起,就可成为真正的游戏程序。下面便介绍两者的连接方法。

▲程序与背景的连接方法

(1)用 BG 绘图程序绘出背景。

(2)按[ESC]键,再按[Pause],回到"菜单"状态。选择 1,进入 BASIC 状态。

(3)键入 BASIC 程序并加入语句。

```
5  CLS ↵                  ;清除背景
键入 RUN ↵                ;试运行不带背景的程序
程序正确以后,键入
5  VIEW ↵                 ;联接背景
键入 RUN ↵                ;带有背景运行程序
```

这时背景和卡通可一同显示出来。

▲几个有关的注意事项

(1)当程序没有 CLS 语句,把 VIEW 语句用在最前面。

(2)当程序有 CLS 语句,则用 VIEW 语句代替 CLS 语句,以便带背景运行。

(3)要修改 BASIC 程序,必须要先消除背景。

(4)用[HOME]可以消除背景。这时光标返回左上角。

(5)如果程序在运行中可用[Pause]键打断。

▲给玛丽沃八方行走加上背景

利用上述方法随便编一点背景，按 ESC 键再按 Pause 键退出 BG 绘图程序。按数字 1 进入 BASIC 状态，写出下面程序：

〈程序 3.10〉

```
 5  VIEW
10  SPRITE ON
20  CGSET 1,0
30  FOR N=0 TO 7
40  DEF MOVE (N)=SPRITE (0,N+1,3,255,0,0)
50  NEXT
60  MOVE 0,1,2,3,4,5,6,7
70  GOTO 60
```

运行上述程序可以发现，刚才编的背景和卡通一起出现在屏幕上。而且，卡通可以重叠在背景之上。

3.12 如何在磁带上存取程序

用 BASIC 所写的程序和用 BG 绘图程序所写的背景，可以用磁带存取。

· 存取步骤有 3 个：

(1)将键盘与计算机相联，并把录音机电源和计算机电源接好。

(2)将键盘读出插口(SAVE 或 WRITE)与录音机(MIC 或 SAVE)插口相联。

(3)将键盘读出插口(LOAD 或 READ)与录音机插口(EAR 或 LOAD)相联。

· 两个注意事项：

音量大小对程序或背景的存取有直接影响，先放在音量的中部试一试，一般需要试几次，才可把音量的大小找准。找准之后应记下音量旋钮的位置，以后每次都在同一音量下进行存取。

对有高音的录音机，应把高音开到最大。

▲用文件方式存取背景画面

文件(文卷)是计算机专业的一外术语，文件内容由数据或程序组成，文件可以有一个文件名，文件名的有效长度为 16 个字符，当有了文件名，存在磁带上的文件在调出时，可以用文件名调出。

· 用 BG 绘图程序存入背景到磁带上

在 3.10 节未介绍的 FILE(文件)功能在这里介绍。

(1)把磁带插入录音机

(2)按 ESC 进入指令菜单

(3)按 ↓ 选择 FILE(文件)指令

(4)按空格键显示出：SAVE(S),LOAD (L)?　　　　;SAVE 表示存入，LOAD 表示调出。

(5)如果按 S 则显示：FILE NAME"　　　　;文件名"

(6)写入文件名 FILE NAME"TEST　　　　;文件名"TEST

(7)按下录音键，再按 RETURN 键，开始存入。

(8)存入完毕，计算机退出 FILE 状态，返回选择(SELECT)状态。

(9)按录音机 STOP 键停止录音。

整个存入过程的提示为：

SAVE (S),LOAD(L)? S　　　　;选择 S

FILE NAME "TEST ;文件名是 TEST

WRITING TEST ;正在写入 TEST 文件

・用 BG 绘图程序从磁带上读出数据

(1)在录音机上插入磁带,磁带上存有用 BG 程序编写的文件。

(2)找到文件的开始部位后停止,把音量和音调调好,当按下录音机放音键后,在有文件的地方,可以听到比较尖锐的声音停机,再次找到文件开始部位,并留上适当空白。按下暂停键和放音键。

(3)按 ESC 键进入指令菜单

(4) ↓ 键选择 FILE(文件)指令

(5)按空格键显示:

SAVE(S),LOAD(L)? ;选择存取状态

(6)按 L 键显示:

FILE NAME" ;请输入文件名

如果这时按了 S 或 L 之外的键,计算机将返回选择状态。

(7)输入要调出文件的名字,按回车键,松开录音机暂停键。

如果没有输入文件名而按回车键,将取出磁带上遇到的第一个文件。

如果在调出文件期间按 Pause 键,返回选择状态。

(8)当取完文件,文件的内容将被显示在屏幕上并返回选择

(9)按录音机停止键(STOP)停止放音。

当错误地读文件或在读文件时按了 Pause 键而中断文件调出,或在屏幕上出现了未完成的图形,这时要用 BG 程序的清除(CLEAR)指令进行清除。

其它错误参见后面的介绍。

整个取出过程的提示为:

SAVE (S),LOAD(L) ;选择 L

FILE NAME "TEST ;文件名是 TEST

SKIP ;跳过其它程序

LOADING TEST ;正在调出名为 TEST 的文件

▲存取 BASIC 程序

・存入 BASIC 程序

存入程序的步骤如下:

(1)在录音机上插入磁带。

(2)作好录音准备,按下录音和暂停键。

(3)键入:SAVE

(4)键入文件名:"TEST"

SAVE"TEST" ;存入名为 TEST 的文件引号不能少。

(5)释放暂停键

(6)按回车键显示:WRITING TEST ;正在写入文件

(7)存入结束显示:

OK

在 OK 的下一行显示光标:

(8)按录音机停止键停止录音

整个存入程序过程的提示为：

SEVE“TEST” ;存入名为 TEST 的文件

WRITING TEST ;正在写入文件

OK ;结束

;光标

(9)要确认存入是否正确(下面将介绍)。

注意：磁带的空白保护带要预先卷过去。

为了正确存入 BASIC 程序，必须要确认是否在 BASIC 状态。若在 BG 程序状态要先退出。

·确认存入程序是否正确的方法

(1)把磁带倒回到欲取出文件的开始部分

(2)键入 LOAD?

(3)按回车键，再按录音机放音键，当磁带转动后屏幕将显示：

LOAD? ;问号不能少

LOADING TEST ;正在调出刚存入的 TEST 文件

(4)取出文件结束后，如果程序存入正确

屏幕将显示：

OK

如果程序存入不正确，则屏幕显示：

LOAD?

LOADING TEST

? TP ERROR ;磁带存取出错

OK

这时请重新存入一次。如果再次存入后依然不能正确调出，请检查录音机状态，或者换一盘磁带再试一试。

·从磁带取出程序

从磁带上取出程序与调出程序检查是不同的。取出程序将使内存中原有的程序破坏。下面介绍取出程序的方法。

取出过程如下：

(1)插入存有程序的磁带，找到磁带开始部位。

(2)键入 LOAD 指令和程序名，即：LOAD“TEST”

(3)按回车键，再按录音机放音键，这时屏幕上显示：LOAD“TEST”LOADING TEST

(4)如果不键入文件名，只用 LOAD 指令并按回车键，这时磁带上遇到的第一个程序被取出。

(5)取出程序结束后屏幕上显示：

LOAD“TEST”

LOADING TEST

OK

如果程序取出有错，屏幕上显示：

LOAD“TEST”

LOADING TEST

? TP ERROR ;磁带存取错误

OK

这时可以再取一下试试，并检查录音机联线、音量等是否合适。

▲出错原因及处理

图 3.7 是整个录音过程的示意。

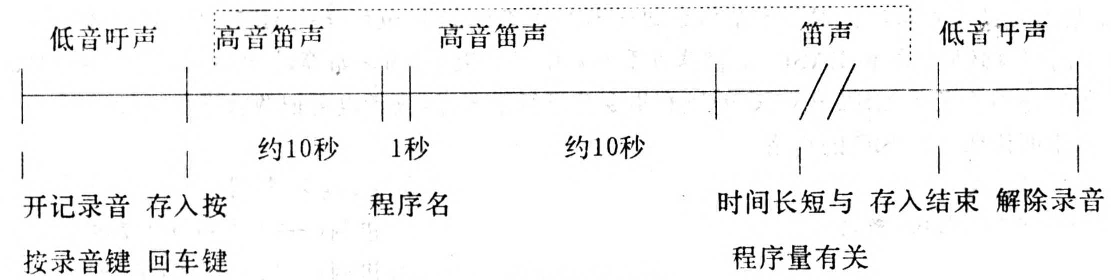


图 3.7 录音过程示意

无论是 BASIC 程序还是 BG 程序的存取，若出现：

? TP ERROR 磁带存取错误

请注意下列事项，再进行存取操作。

无法正确存入程序(文件)

(1)键盘与录音机联接错误

(2)磁带与录音机不匹配

(3)磁带上有疵点

(4)录音机走带不准(不匀)

(5)外部杂音干扰

(6)录音机不适于录入文件

处置建议

(1)正确联线

(2)更换磁带

(3)更换录音机

(4)让录音机远离电视机，去掉录音机上多余的联线。

无法正确调出

在能够正确存入不能正确调出时

(1)录音机音量不适合，或联线错误。

(2)有杂音进入。

(3)没有找到文件的起始点。

处置建议

(1)调整录音机的音量

(2)确认元件的起始点

(3)远离电视避开干扰

四、FBASIC 语法篇

4.1 FBASIC 规格说明

在本章将系统地叙述第三章中用过的指令，以及其它指令。FBASIC 对指令的规定与 MSX BASIC 大致相同。增加了部分专用指令扩大了 FBASIC 处理游戏的能力，压缩了部分数值计算功能，使 FBASIC 处理数值的能力比 MSX BASIC 弱。MSX BASIC 是日本为家庭计算机(8 位计算机)所制定的工业标准。对于本章指令叙述不太理解的读者，也可寻找 MSX BASIC 语言作为参考书。例如，《微型计算机 BASIC 语言速查手册》(中村八束编、杨孝如等译，电子工业出版社，1987 年出版)，该书收录了 MSX BASIC 的所有指令并进行了解释，读者也可把两者进行比较。

下面说明 FBASIC 的规

字符种类：	数字、英文、符号
数字表示范围(整数)：	10 进制(—32768～＋32768) 16 进制(&H0000～&HFFFF)
字符串范围	0～31 个
变量名：	255 个字符，只识别前两位
行号范围：	0～65535
行的最大容量：	255 位
多个语句：	一行内多个语句用冒号隔开
编辑功能：	全屏幕编辑
画面状态：	4 层
画面密度	背景面(28 字×24 行) 卡通面(256×240 点) 字：8×8 点阵
彩色：	用彩色发生器产生 52 色
音乐功能：	音符、拍节、三重和音、音色
操纵器：	1 号、2 号操纵器
通信：	1200 波特
指令数：	基本指令 74 条
卡通图案设定：	从已规定的 16 种中选

在介绍每个指令前，先介绍一下有关的预备知识

4.2 运算符号

▲运算符

运算符可以分成三类：

(1)算术运算符

其它未规定的运算符比如乘方，是无效的。表 4.1 是算术运算符的示意。

表 4.1　算术运算符说明

算术运符	运算符含义	示例	对应的数学表达式	
+	加	A+B	A+B	相同
−	减	A−B	A−B	相同
*	乘	A*B	A×B	不同
/	除	A/B	A÷B 或 A/B	基本相同
MOD	取余数	A MOD B		

取余数的意思是：

若 10÷3=3，余数为 1。

用 MOD 表示为：

10MOD3，结果为 1。

算术运算符的运算优先顺序分为三级：

①乘、除（*、/）

②取余数（MOD）

③加、减（+、−）

当优先级相等，将从左向右执行表达式。

所谓表达式，是用变量、常数和运算符号组成的表达特定数学含义的式子。在计算机中还要考虑表达式是否合法（符合规则）。不合规则的表达式，计算机将会不识别。

为了改变优先顺序可以采用小括弧。

(A+B)*Y

这时先算 A+B，再算乘 Y。

(2)**关系运算符**

关系运算符是用于比较两个常数或两个变量的符号。

关系运算符有：

关系符	含义
=	等于
<>	不等于
>	左边大于右边
<	左边小于右边
>=	左边大于等于右边
<=	左边小于等于右边

关系符在条件语句 IF～THEN 中有应用，下面便是一例：

IF　X>0　THEN　1000

意思是 X 的值若大于 0，则跳到第 1000 行。

下面是一判断语句

A=X>0　　或者　A=(X>0)

对于上式：若 X>0 则 A 被代入 −1

若 X<0 或 X=0 则让 A=0

由 A 的值可以判断 X 是否大于 0。

(3)**逻辑运算符**

逻辑运算符用于判定真伪逻辑关系。当判定为真时得1,当判定这伪时得0。逻辑运算的特点,决定了它常常用于二进制的运算中。下表是逻辑符及其表示的含义。

逻辑符	含义
NOT	非
AND	与
OR	或
XOR	异或

表4.2用X和Y变量为例解释了逻辑运算符的含义。

在IF～THEN语句中可以用逻辑运算符来改变程序流程:

```
IF X>0 AND X<10 THEN 1000
```

意思是,当X>0与X<0同时成立,则跳到第1000行。这一语句等效下面两条语句:

```
10  IF X≤0 THEN 30          ;X≤0条件成立则跳转30行
20  IFX<10 THEN 1000        ;X<10条件成立则跳到1000行
30……
```

表4.2 逻辑运算符示例

逻辑符	变量	逻辑符+变量	解释
NOT(非)	X 1 0	NOT X 0 1	当X为1 NOT X为0 当X为0 NOT X为1 即NOT X和X是相反的。
AND(与)	X Y 1 1 1 0 0 1 0 0	X AND Y 1 0 0 0	X AND Y表示逻辑乘,当X和Y为1,结果为1,否则为0。
OR(或)	X Y 1 1 1 0 0 1 0 0	X OR Y 1 1 1 0	X OR Y表示逻辑加,但是1+1不等2而是等于1。
XOR(异或)	X Y 1 1 1 0 0 1 0 0	X XOR Y 0 1 1 0	X XOR Y表示X和Y是相同值时,结果为0,否则为1。

用逻辑运算符,必须注意逻辑关系是否成立,下面的例子是错误的:

```
IF X<0 AND X>10 THEN 1000
```

意思是当X<0与>10成立让程序跳转到第1000行。

上面已经介绍了三种运算符号,下面把运算符号的优先顺序归纳如下:

(1) 括号 ( )

(2) 函数

(3) 乘除 *、/

(4) 取余 MOD

(5) 加减 +、-

(6) 等于 =、不等于 <>、大于 >、大于等于 >=、小于等于 <=

(7) 非 NOT

(8) 与 AND

(9) 或 OR

(10) 异或 XOR

▲字符变量的逻辑加(逻辑与)

字符变量相加不同于数字相加,字符变量相加是比较常用的。下面的写法是合法的。

```
C$=A$+B$
C$="字符串"+B$
C$=C$+"字符串"
```

字符变量相加是把字符变量的值连在一起,下面的程序可说明这一点。

```
10 INPUT  A$
20 INPUT  B$
30 C$="CCC"
40 D$=A$+B$
50 C$=D$+C$
60 PRINT  D$,C$
RUN
? AAA                    ;输入字符到 A$
? BBB                ;输入字符到 B$
AAABBB  AAABBBCCC    ;运行结果
```

▲几个具有特殊用途的标点符号

(1)—连接号

在 LIST 指令中,可以用连接号把任意一段程序调出来。如:

```
LIST  100—300↓
```

(2),逗号

在输入语句 INPUT、打印语句 PRINT、数据语句 DATA 等中,如有多个操作数时,每个操作数中间用逗号隔开。有时人们把指令如 INPUT、DATA 等称为操作符,而把这些指令所带的参数,如变量、常数称为操作数。下面是逗号的应用:

```
INPUT  A,B,C
DATA  10,20,30,40
PRINT  A,B
```

(3):冒号

在同一行号下写入一条以上语句时,两条语句之间要用冒号隔开。如:

```
A=B-C:PRINT A
```

(4);分号

在 INPUT 及 PRINT 语句中，当多于一个操作数时，两个操作数之间可以用分号隔开。用分号隔开与用逗号隔开不同，用分号隔开变量，在打印语句中，结果之间仅空一格，而用逗号隔开则空 7 格，下面是分号与逗号隔开的例子：

```
10  INPUT   A,B
20  PRINT   A;B                    ;打印语句的作用是把结果显示在屏幕上。
30  PRINT   A,B
```

(5)? 问号

用问号代替 PRINT，以节约空间。如：

? A,B,C 等于 PRINTA,B,C

(6)&H××表示 16 进制数，如：

&H2C 即 10 进制 44

▲二进制、十六进制、十进制比较

关于二进制、十六进制的讲解书籍已经很多，这里仅举一例加以说明。

所谓十进制就是逢十进一，而二进制呢？是逢二进一，而十六进制就是逢十六进一。二进制和十六进制是为了计算机计算上的方便，对于用惯十进制的人，自然不太方便。下面比较一下三种进制的特点，如表 4.3 所示。

十进制数 255 用十六进制表示为 FF；用二进制表示为 11111111。

本章提供了十六进制和十进制数之间的转换指令。

4.3 语法部分的学习方法

在解释每一个指令之前，先介绍本书描叙每个指令的方法。

功　能：解释每个指令的功能

格　式：说明指令构成语句的书写格式。格式中采用的符号和字母遵照下述规定。

(1) 英文大写字母可直接键入；

(2) 中括孤〔〕内的内容可任意省略，但中括孤不要输入到语句中；

(3) 大括弧{　}内的内容可任选一项；

(4) 括号、逗号、冒号、分号、连接号等符号，必须用在所指定的位置；

(5) 省略号……表示在一行(255 个字符内)范围内可重复多次相同或有规律的内容；

(6) 间隔符␣表示在两个字符之间有间隔；

(7) 引号“　”表示其中的内容是字符串；

(8) 其它有关语法内容在下面进行叙述。

表 4.3　三种进制的比较

十　进　制	二　进　制	十　六　进　制
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	10000	10

缩　写:指令的缩写符号。缩写符号计算机可以识别。

解　释:详细说明指令功能或注意事项。

程序例:适当的例子或执行结果。由于程序可能在一行内显示不下,这时要不改行号,连续把一行程序输入完。

缺省值:如果没有给指令指定参数,缺省值作为参数将自动地被处理。

4.4　系统实用指令

CLEAR

功　能:设置用户程序使用的内存范围。

格　式:CLEAR &H××

内存位置通常用 16 进制数表示。"&H"被用在 16 进制数前。

缩　写:CLE

解　释:为了防止程序过大进入系统控制程序区,用 CLEAR &H××指令可以预先设置程序的范围,以避免程序丢失。

仅使用 CLEAR,可以清除内存所有变量以得到更多的空间。

程序例:10　REM ＊ CLEAR ＊　　　;这是一条注释语句,后面将介绍。

20　CLEAR &H7600　　　;设定程序只能在十六进制 7600 单元之前的区域内。

NEW

功　能:清除内存区中所有程序和变量。

格　式:NEW

缩　写:

解　释:执行 NEW 指令将清除内存中已有的程序(用户自己输入的程序)和使用的变量。

LIST

功　能:把内存中已输入的程序按行号大小,在屏幕上列出。即列出程序清单。

格　式:LIST　[m][-[n]]

　　m→表示起始行号

　　n→表示结束行号

缩　写:L.

解　释:LIST 指令的作用是列程序清单。当程序被在屏幕上列出后,可以较容易的发现错误从而有利于修改。

利用 LIST 指令,可以任意地列出所想看的程序。说明如下

LIST m	;只列出第 m 行程序
LIST m—	;只列出从 m 行到结束行的程序
LIST m—n	;列出从 m 行开始到 n 行结束的程序
LIST	;列出所有程序

程序例:

```
10  REM * LIST *   ┐
20  INPUT X        │
30  Y=X+Y          ├输入的程序
40  PRINT X;Y      │
50  END            ┘
LIST 30            ┐
30  Y=X+X          ├仅列出 30 行
OK                 ┘
LIST 40—           ┐
40  PRINT X;Y      │
50  END            ├列出 40 行之后的程序
OK                 ┘
LIST —20           ┐
10  REM * LIST *   │
20  INPUT X        ├列出 20 行之前的程序
OK                 ┘
LIST               ┐
10  REM * LIST *   │
20  INPUT X        │
30  Y=X+X          ├列出所有程序
40  PRINT X;Y      │
50  END            │
OK                 ┘
```

RUN

功　能:运行内存中的程序

格　式:RUN[m]

　　m→表示开始行号

缩　写:R.

解　释:(1)如果仅键入 RUN 指令,程序从开始执行。

(2)如果键入 RUN m，表示从第 m 行开始执行。利用这种方法可以检查程序某一部分是否有错误。

(3)关于程序运行期间的打断和继续进行，请参看对 STOP 指令和 CONT 指令的叙述。

程序例：

```
10  REM * RUN *          ┐
20  PRINT "FAMILY"       │
30  PRINT"BASIC"         ├输入的程序
40  END                  ┘
    RUN                 ;运行程序
        FAMILY ┐
        BASIC  ├;显示程序运行结果
        OK     ┘
```

STOP

功　能：中断程序运行

格　式：STOP

缩　写：STO.

说　明：在调试程序时，STOP 指令是很有用的。当程序执行到某一段时，插入一个 STOP 指令把程序中断掉，看看程序运行的中间结果是否和预想的一致？若没有问题，再用 CONT 指令继续运行。在键盘上有一个 Pause 键，产生的中断效果同 STOP 指令。也可用 CONT 指令继续运行。

程序例：

```
10  REM * STOP *
20  FOR I=1 TO 10 ┐
30  PRINT I       │
40  STOP          ├键入的程序
50  NEXT          ┘
RUN
1                                  ;显示结果 1
BREAK IN 40                        ;显示在第 40 行中断
OK
CONT                               ;键入 CONT
2                                  ;显示结果 2
BREAK IN 40                        ;显示在第 40 行中断
OK
```

CONT

功　能：令被中断的程序继续执行

格　式：CONT

缩　写：C.

说　明：(1)用 Pause 键或 STOP 指令产生的程序中断，可以用 CONT 指令继续程序的执行。

(2)在程序执行中产生出错信息而程序也被中断时，CONT 指令可以从出错的下一行继续

执行。

程序例:参见 STOP 一节 END

功　能:使程序终止执行

缩　写:E.

说　明:END 指令也可用于判断语句中,而并让程序终止。通常 END 指令可被用在程序的最后一行。在有子程序的程序中 END 所在的行代表结束行。

程序例:

```
10  REM * END *              ;执行这个程序将依次打印出 1,2,3……假如按下 Z
20  FOR I=1 TO 1000          ;键,程序将结束。否则,将依次打印到 1000
25  PRINT I;                 ;
30  A$=INKEY$                ;
40  IF A$="Z"THEN END        ;
50  NEXT                     ;
RUN                          ;
     1 2 3 4 5 6…            ;结果
```

LOAD

功　能:将磁带上的文件输入到内存中。

格　式:LOAD["文件名"]

缩　写:LO.

解　释:(1)文件名是由 SAVE 指令所指定的文件名。

(2)执行 LOAD 指令将清除内存原有的程序和变量

程序例:

```
LOAD                  ;取出磁带上的第一个文件
LOADING TEST          ;文件名为 TEST
OK                    ;
LOAD "TEST—2"         ;取出名为"TEST—2"的文件
SKIP TEST             ;跳过名为 TEST 的文件
SKIP TEST—1           ;跳过名为 TEST—1 的文件
LOADING TEST—2        ;正在把名为 TEST—2 的文件调出
OK                    ;调出结束
```

SAVE

功　能:把内存的内容存入磁带上。

格　式:SAVE["文件名"]

缩　写:SA.

解　释:(1)内存中的程序或数据,断电后一般不能保存,需要保存的程序和数据可以存在磁带上,以便多 次使用。

(2)每存入程序或数据都应起一个文件名,当用户想使用某一个程序,只需要知道文件名,使用 LOAD 指令可把该文件调入内存中。

程序例:

```
SAVE                ;存入文件无名
WRITING             ;正在存入
OK                  ;存入结束
SAVE"TEST"          ;存入名为 TEST 的文件
WRITING TEST        ;正在存入
OK                  ;存入结束
```

LOAD?

功　能:检查所存入的程序是否正确,并能否正确调出。

格　式:LOAD? ["文件名"]

缩　写:LO. ? 或 LO. P

解　释:(1)用 LOAD? 检查磁带上的程序和内存中的程序是否相同。不指定文件名将调出最后存入的程序进行核对。

(2)若指定了文件名(16 个字符内),计算机将跳过其它程序,找到所指定的文件。用 LOAD? 检查存在磁带上的程序十分重要,在存入程序时不要漏过这一步。

程序例:

```
LOAD?               ;取出最后存入的程序
LOADING TEST        ;正在核对
OK                  ;存入正确,结束
LOAD? "TEST"        ;取出名为 TEST 的文件
LOADING TEST        ;正在核对
OK                  ;存入正确,结束
? TP ERROR          ;存入错误(调出错误)
OK                  ;
```

4.5 基本指令

赋　值

功　能:给变量赋值

格　式:变量=[字母、常数、字符串或语句]

解　释:(1)赋值是把等号右边的值赋给左边。左边必须是变量。

(2)在等号右边 的变量必须预先被定义。未被定义的数值变量将自动地赋给 0 值。未被定义的字符变量被自动赋给"空格"。

(3)等式两边变量类型必须相等,即当左侧是数值变量右侧也是数值的;当左侧是字符变量右侧也是字符的。如:

正　　确	错　　误
X=A	A $ =A
Y=A * X	Y $ =A * X+"10"
X $ ="&H"+A $	X="&H"+A $

程序例:

```
10  REM * LET *
```

```
20  A=10                          ;数值变量被赋值 10
30  A$="ABC"                      ;字符串变量被赋值 ABC
40  PRINT A$;A                    ;打印变量 A$和 A 的值
RUN                               ;
ABC 10                            ;得出的结果
```

PRINT

功　能:将程序运算结果在屏幕上打印出来。

格　式:

PRINT["字符串"] {;/,} [常数/变量/表达式/字符串] [{;/,} [常数/变量/表达式/字符串] …]

缩　写:? 或 P.

解　释:用 PRINP 可以在屏幕上打印出程序运行的结果,也可打印字符串、变量、常数等。在变量、字符串或常数之间插入分号,可以依次打印;在变量、字符串或常数之间插入逗号可按格式打印。

INPUT

功　能:在程序运行期间从键盘键入数字数据。

格　式:INPUT〔"字符串?〕{;/,}变量〔,变量,…〕

缩　写:I.

解　释:(1)当程序执行 INPUT 指令,在屏幕上打印问号等待从键盘上输入数据,输入数据后执行下一个指令。

(2)字符串是一串字符,它将被显示在屏幕上,在其之后显示问号。

(3)变量分成数值变量和字符变量,在 INPUT 指令中,数值变量只接受数字(0～9)组成的数。字符变量可以接受除逗号外的所有字符。

(4)假如变量多于一个,必须用逗号把变量分开,而输入数据也须用逗号分开以区别输入到不同的变量。否则回车之后将出现两个问号,等待给下一个变量送数。

程序例:

```
10  REM * INPUT *                 ;
20  INPUT"A=";A                   ;请输入数据到 A
30  INPUT"B=";B                   ;请输入数据到 B
40  C=A+B                         ;C 等于 A+B
50  D=A-B                         ;A、B、C 的数值在-32768～+32768 之间
60  E=A*B                         ;
70  PRINT"A+B";C                  ;打印 A+B
80  PRINT"A-B";D                  ;
90  PRINT"A*B";E                  ;
```

LINPUT

功　能：输入字符串

格　式：LINPUT〔"字符串"〕$\left\{\begin{matrix};\\,\end{matrix}\right\}$字符变量

解　释：(1)LINPUT 与 INPUT 不同之处在于可以输入包括逗号在内的符号。可输入的字符数最多为 31 个。仅可以指定一个字符变量。

(2)LINPUT 指令中只要用了提示用的字符串，尽管在打印指令中不在指定也可以打出用于提示的字符串，所以在必要时需把提示除去。

(3)LINPUT 指令是把提示字符串与字符变量进行了相加处理，即等效为：

A＄＝"字符串"＋A＄

(4)执行 LINPUT 指令在屏幕上不出现问号。

程序例：
```
10  LINPUT"STRING＝";A＄
20  PRINT A＄
RUN
STRING＝A2,3                    ;输入数据，"STRING＝"是提示。
STRING＝A2,3                    ;打印结果。
```

CLEAR

功　能：清除内存中所有变量的值。

格　式：CLEAR

缩　写：CLE

解　释：数字变量清除之后变为 0，字符变量变为空。

程序例：
```
10  X＄＝"ABC"                  ;字符串变量值为 ABC
20  Y＝178                      ;让 Y＝178
30  PRINT X＄;Y                 ;打印 X＄,Y 的值
RUN                             ;运行
ABC 178
OK
CLEAR                           ;执行清除指令
RUN                             ;运行
0                               ;Y＝0  X＄＝空
OK                              ;
```

DIM

功　能：定义一个数据矩阵或数组，并在内存开辟数据区域。

格　式：DIM 变量名〔ml〔,m2〕〕〔,变量名 nl〔,n2〕〕

缩　写：DI.

解　释：DIM 定义的数值和矩阵可以是数值型的，也可以是字符型的。所定义数组和矩阵的大小受到内存容量的限制。一般数组可达 X(255)，矩阵可达 B(43,43)，当定义的数组多时，每个数组还要小。

程序例：(1)定义数值变量

```
10  REM * DIM——(1) *
20  DIM A(3),B(3,3)
30  FOR I=0 TO 3  ┐
40  A(I)=I        │
50  PRINT A(I);   │ ;把 0～3 存入 A(I)中
60  NEXT          ┘
70  PRINT :PRINT
80  FOR I=0 TO 3    ┐
90  FOR J=0 TO 3    │
100 B(I,J)=I* 10+J  │ ;给 B(I,J)存入 4×4 的数字矩阵
110 PRINT B(I,J);   │
120 NEXT            │
130 NEXT            ┘
```

(2)定义字符串变量

```
10  REM * DIM———(2) *
20  DIM A$(3),B$(3,3)
30  FOR I=0 TO 3
40  A$(I)="TEST"+STR$(I)               ;STR$为字符函数
50  PRINT A$(I)
60  NEXT
70  PRINT
80  FOR I=0 TO 3
90  FOR J=0 TO 3
100  B$(I,J)="TOKYO"+STR$(I)+STR$(J)
110  PRINT B$(I,J)
120  NEXT
130  NEXT
```

GOTO

功　能:指定程序跳转的行号。

格　式:GOTO N

　　　　N→表示行号

缩　写:G.

解　释:用以改变程序的流向,如使用 GOTO 20,程序跳到行号为 20 的语句开始执行。

程序例:

```
10  REM * GOTO *
20  A$=INKEY $(0)             ;任意按一键程序继续
30  PRINT A$
40  IF A$="Z"THEN END         ;按下 Z 键程序结束,否则继续
50  GOTO 20                   ;再跳回 20 行重行执行
```

GOSUB

功　能:转入指定行号的子程序继续执行。子程序的结尾必须用 RETURN 指令返回。

格　式:GOSUB　N

缩　写:GOS.

解　释:当一部分程序被频繁使用,若反复写多次,将增加程序的长度。这时可把频繁使用的那一段程序单独提出来作为子程序,每次需要执行那段程序时用 GOSUB 指令调用,而被调用的子程序必须用 RETURN 返回。一般子程序写在程序的后面。

程序例:

```
10  REM * GOSUB *                        ;10—170 是主程序
100 FOR I=1 TO 25
110 GOSUB 1000                           ;调子程序
120 NEXT
130 FOR I=25 TO 1 STEP -1                ;循环语句,但步长为-1。
140 GOSUB 1000                           ;调子程序
150 NEXT
160 PRINT"END"
170 END
1000 FOR J=0 TO I                        ;1000—1040 是子程序
1010 PRINT " * "
1020 NEXT
1030 PRINT
1040 RETURN
```

RETURN

功　能:从子程序返回。

格　式:RETURN[N]

N→指定返回的行号,可以不指定。

缩　写:RE.

解　释:RETURN 和 GOSUB 联合使用。当不指定行号,RETURN 指令一被执行,返回 GOSUB 语句的下一个行号并继续执行主程序。

程序例:参看 GOSUB 一节

IF～THEN(条件指令)

功　能:根据条件式的值确定程序的转向。

格　式:IF 条件式 THEN $\begin{Bmatrix}\text{行号}\\\text{语句}\end{Bmatrix}$

缩　写:IF T.

解　释:(1)条件式可以是算术、字符串或逻辑表达式。行号是所要跳转的行号。语句是一条合法的BASIC 语句供执行用。

(2)IF 与 THEN 必须配对出现。当条件式成立则执行 THEN 后的内容。如:

IF X>0 THEN 100

这一语句的意思是假如 X 大于零,则跳转到第 100 行,否则顺序执行。

又如:

IF A $ ="Y"THEN PRINT "GOOD"

这一语句的意思是，若 A$="Y"，则打印出 GOOD 来。

程序例：

```
10  REM * IF—THEN *                      ;
20  PRINT"PUSH Y!";                      ;打印 PUSH Y!，请按 Y。
30  A$=INKEY$(0)
40  IF A$<>"Y"THEN BEEP:                 ;BEEP 指令被执行将发出"嘀"声。
    GOTO 30
50  PRINT :PRINT"PRESS Y"                ;
60  PRINT                                ;
70  GOTO 20
```

行号 30 是准备从键盘上接收字符。

行号 40 是判断所键入的是否是 Y，如果不是发出"嘀"的一声，跳回 30 行准备再次接收。若是 Y，则顺序执行第 50 行。

FOR～NEXT(循环指令)

功　能：反复执行 FOR 与 NEXT 之间的程序行。

格　式：FOR 变量=初始值 TO 结束行 STEP 步长……

NEXT

解　释：(1)变量为数值变量称为循环变量，初始值、结束值和步长均为整数，步长可正可负。省略 STEP 步长时，步长为 1。

(2)FOR 表示循环开始，NEXT 表示循环结束，两者需配对出现。例如：

```
┌FOR I=1 TO 10 STEP 2
│[处理程序]
└NEXT
```

当程序执行到 FOR 语句时，先把循环变量 I 置为 1，执行下面的处理程序，到 NEXT 语句返回 FOR。这时循环变量按 STEP 中所规定的步长增加，如 STEP2 表示步长为 2，这时循环变量的值由 1 增到 3，然后再执行一遍处理程序。如此循环，循环变量依次为 5，7，9 当大于 10 时，循环结束。执行 NEXT 的下一行语句。

(3)FOR 与 NEXT 必须配对出现，并且 NEXT 之后不可加循环变量，这一点希引起注意。

```
┌FOR I=1 TO 10              ;三重循环，先内后外
│┌FOR J=1 TO 20             ;在 NEXT 之外，不可加变量
││┌FOR K=1 TO 5
│││ ⋮
││└NEXT
│└NEXT
└NEXT
```

程序例：

```
10  REM * FOR—NEXT(1) *
20  FOR I=0 TO 10 STEP 2
30  PRINT I;
40  NEXT                   ;打印循环变量的值
RUN
```

```
0  2  4  6  8  10
OK
10  REM * FOR-NEXT(2) *
20  CLS
30  FOR I=1 TO 20
40  FOR J=1 TO I        ;三重循环
50  FOR K=1 TO J
60  PRINT"X"
70  NEXT                ;用 X 建立图形
80  PRINT
90  NEXT                ;PRINT 并不是可有可无,起到换行作用
100  PRINT
110  NEXT
```

ON

功　能:通过变量控制执行不同的程序。即选择执行功能。

格　式:ON 变量 {GOTO / GOSUB / RETURN / RESTORE} 〔行号,行号……〕

缩　写:O.

解　释:下面试举一例说明 ON 指令的功能。由下例可知,原来用 5 条语句,现在只写一条就够了。

```
IF X=1 THEN 1000
IF X=2 THEN 2000
IF X=3 THEN 3000
IF X=4 THEN 4000
IF X=5 THEN 5000
        ↓  等效于
ON X GOTO 1000,2000,3000,4000,5000
```

图 4.1 表示了上述语句的含义。

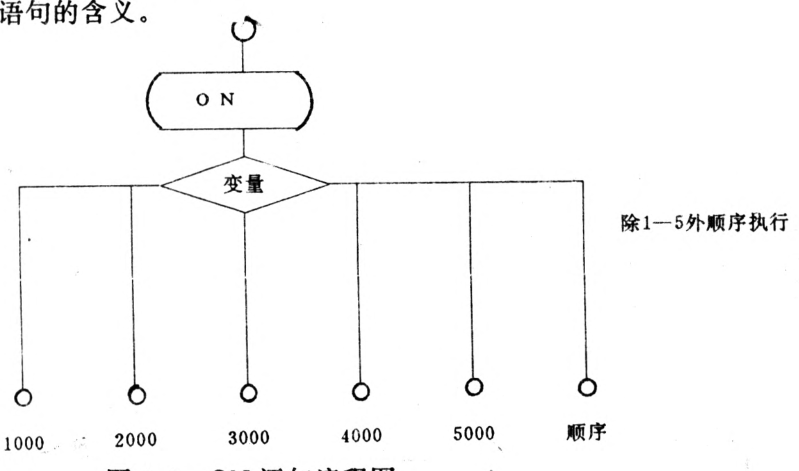


图 4.1　ON 语句流程图

程序例:

```
10  REM * ON—GOSUB *
20  INPUT"KEY IN 1—6";N          ;20 行输入 N 值 1—6
30  ON N GOSUB 100,200,300,400,500,600;30 行是 ON 语句,在 N 为 1—6 以内转入子程序
40  IF N<1 OR N>6 THEN 20        ;40 行条件跳转,N 在 1—6 之外,转回 20 行。
50  PRINT N;"—";X$               ;50 行是打印语句,打印 X$ 的值。
60  GOTO 20
100  X$="HOLLO 1":RETURN         ;100—600 是子程序
200  X$="HOLLO 2":RETURN
300  X$="HOLLO 3":RETURN
400  X$="HOLLO 4":RETURN
500  X$="HOLLO 5":RETURN
600  X$="HOLLO 6":RETURN
```

SWAP

功　能:2 个变量值交换

格　式:SWAP 变量,变量

缩　写:.

解　释:可将两个变量的内容进行交换,但是数值变量和字符变量的内容不能互换,因为两个变量的类型不同。

程序例:

```
10  REM * SWAP *
20  DIM A(10)
30  FOR I=1 TO 10
40  READ A(I)
50  PRINT A(I);
60  NEXT
70  FOR I=1 TO 10
80  FOR J=1 TO 10
90  IF A(1)<A(J) THEN SWAP A(I),A(J)
110  NEXT
110  NEXT
120  PRINT
130  FOR I=1 TO 10
140  PRINT A(I);
150  NEXT
160  DATA 2,3,5,1,7,4,8,9,6,0
RUN
    2  3  5  1  7  4  8  9  6  0
    0  1  2  3  4  5  6  7  8  9
OK
```

40 行和 160 行是读数据语句,下面将介绍。这两行语句的目的是把 160 行中的数据读入 A

(I)中。

第 50 行打印的结果为交换之前的 A(I)。

第 90 行为交换行，通过条件判定可把数字大小按次序重新排在 A(I)中。

第 140 行为打印语句，可把 A(I)中经过交换的值打印出来。

REM

功　能：注解

格　式：REM〔注释内容〕　注释内容不超过 255 个字符。

缩　写：'

解　释：REM 之后的内容不执行仅供注释之用。可以随意键入程序说明，使用日期等内容。

程序例：

```
10  ' * REM *                  ;用'号代替 REM
20  REM FBASIC V1.0
30  REM 1989.6.1
RUN                            ;运行后什么都没有显示
OK
```

READ

功　能：把写在 DATA 指令中的数据读出来。

格　式：READ 变量〔，变量，变量，……〕

缩　写：REA.

解　释：有读指令 READ，就一定要有数据指令 DATA。而数据的个数，一般不能少于读的次数，但是，如果用 RESTORE 指令重新设置了数据的指针，可使数据被重读。

程序例：(1)用 X 作为变量读数据

```
10  REM * READ———(1) *
20  FORI=1 TO 10
30  READ X                     ;注意：读入到变量 X 的值不能被保留，
40  PRINT X;                   ;因为 X 的值每次都将被新读入的值所
50  NEXT                       ;代替，最后 X 中保留的值是 9。
60  DATA 3,4,1,6,2,7,8,3,4,9
RUN
3  4  1  6  2  7  8  3  4  9
OK
```

(2)用字符串变量读数据

```
10  REM * READ———(2) *
20  READ A$,B$,C$
30  PRINT A$;" ";B$;"."
40  PRINT A$;" ";C$;"."
50  DATA GOOD,MORNING,EVENING      ;三个数据分别被读入 A$,B$,C$
RUN
GOOD MORNING.
GOOD EVENING.
OK
```

(3)用数组变量读数据

```
10  REM * READ———(3) *
```

```
20  DIM A(5)
30  FOR I=0 TO 5
40  READ A(I)                    ;数据被读入 A(I)中保存起来
50  PRINT A(I);
60  NEXT
70  DATA 9,1,8,3,4,8
RUN
9  1  8  3  4  8
OK
```

DATA

功　能:定义读指令所需的数据。

格　式:DATA 常数〔,常数,常数,……〕

缩　写:D.

解　释:常数分为数值常数和字符常数(字符串),数据指令和 READ 指令配合使用。符号可以放在引号内。如:

DATA ABC,DE,“,”,F

逗号放在引号中了。

RESTORE

功　能:把数据 DATA 中的指针指向第一个数据,使 DATA 的数据可以重新使用。

格　式:RESTORE〔行号〕

行号→DATA 所在的行号

缩　写:RES.

解　释:当有一些数据有重复使用的情况,这时用 RESTORE 设置重复部分的指针,使重复的数据可以用读指令重读。

程序例:
```
10  REM * RESTORE *
20  RESTORE 1010                 ;20 行把数据指针设在 1010 行
30  FOR I=0 TO 5
40  READ A                       ;40 行把 1010 行数据读出
50  PRINT A;
60  NEXT
70  PRINT
80  RESTORE 1000                 ;80 行把数据指针设在 1000 行
90  FOR I=0 TO 5
100  READ A                      ;100 行把数据读出来
110  PRINT A;
120  NEXT
130  REM
1000  DATA 23,43,55,65,42,9
1010  DATA 12,56,34,68,53,2
RUN
12  56  34  68  53  2            ;打印出的结果
23  43  55  65  42  9
OK
```

POKE

格　式:可直接把数据放入内存单元。

格　式:POKE 内存单元地址,欲放入内存单元的数,数→在 0～255 范围内。

缩　写:PO.

解　释:这是一个比较难理解的指令。尤其是对于没有系统地学习过计算机知识的读者,不太明白这个指令的作用。这个指令使人们可以直接修改内存的数据,是很有用的一条指令。

PEEK

功　能:PEEK 是一个函数,用于读出内存单元的数据。

格　式:PEEK(变量)

缩　写:PE.

解　释:这个指令仅用于读出内存单元的数据而不改变内存单元的值。

程序例:

```
10  REM * POKE *
20  CLEAR &H7600
30  D=0
40  FORA=&H7600 TO &H761F       ;40～70 行是在内存 7600 到
50  POKE A,D                    ;761F 单元分别放入 0～31
60  D=D+1
70  NEXT
80  FOR A=&H7600 TO &H761F      ;80～110 行是把 7600～761F 单元的数据读出,
90  RD=PEEK(A)                  ;并用 16 进制的形式显示出来。
100  PRINT"  ";HEX $ (RD);      ;HEX $ (RD)便是表示 16 进制数的指令,后面将介绍。
110  NEXT
```

4.6　屏幕控制指令

LOCATE

功　能:指定数据在屏幕上的位置。

格　式:LOCATE X,Y

　　X→水平方向位置 0～27

　　Y→垂直方向位置 0～23

缩　写:LOC

解　释:LOCATE 是用于在背景面定位的指令。屏幕如图 4.2 所示。利用 LOCATE 可以改变光标在屏幕上的位置。

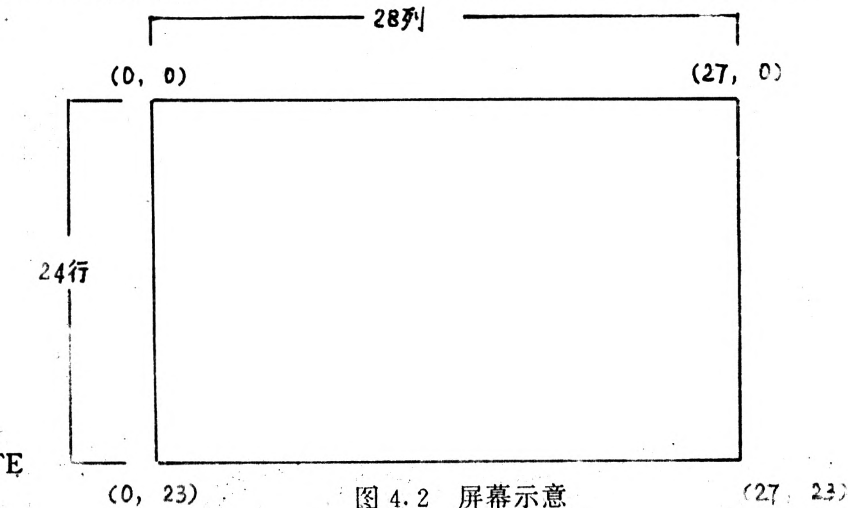


图 4.2　屏幕示意

程序例:

```
10  REM * LOCATE
20  CLS
30  FOR I=0 TO 20
40  LOCATE I,I:PRINT"*";        ;第 40 行利用变量 I 把 * 打印在不同位置上
```

50　NEXT
60　LOCATE 0,10　　　　　;程序最后把光标定位在(0,10)位置上

COLOR

功　能:让背景面某个位置显示某种颜色。

格　式:COLORX,Y,n

X→水平方向座标 0～27

Y→垂直方向座标 0～23

n→配色代码 0～3

缩写:COL.

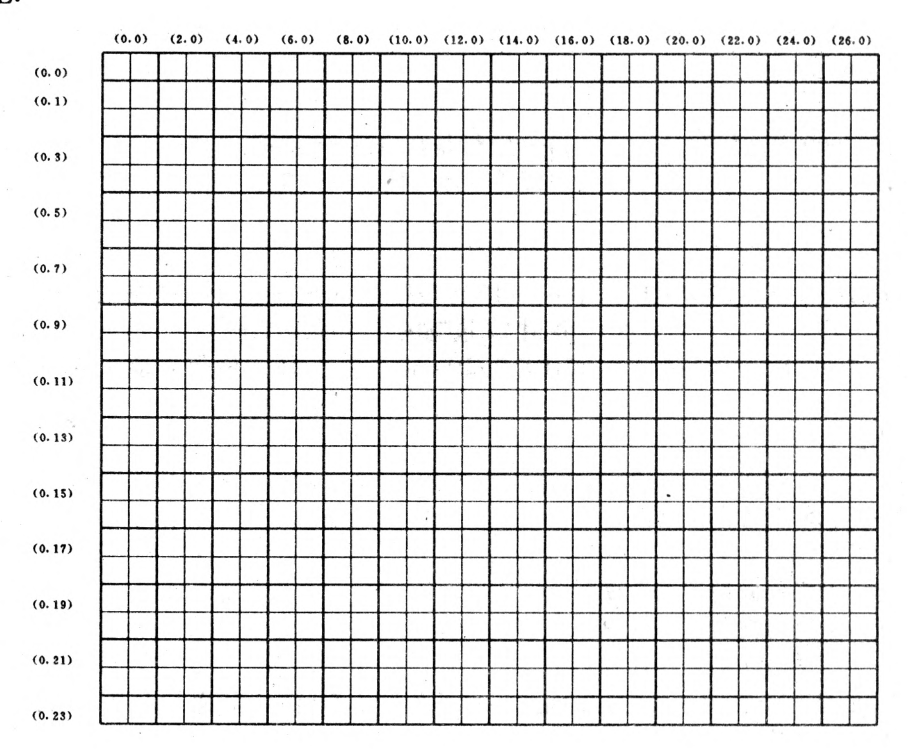


图 4.3　屏幕可分成若干个 16×16 方块

解　释:(1)图 4.3 表示了一个屏幕,可按 4 格组成一个个方块。利用 COLOR 指令指定 X,Y 座标,可把方块的内部按配色代码加以配色。背景用配色代码的含义参见封 3 和表 4.6。

A、B、C、D 四块组成 16×16 点阵即四个字符大小

	10	11	
第 9 行	A	B	
第 10 行	D	C	

例如:COLOR　10,10,3 可把座标(10,10)所在的方块中的其余三块 A,B,C 按配色代码 3 配色。

(2)COLOR 只改变背景面的颜色,在使用 COLOR 之前,要注意 CGSET 中所选择的板代码是什么。

程序例:10　REM * COLOR *

```
20  CLS
30  FOR I=0 TO 447
40  PRINT CHR$(195);
50  NEXT
60  FOR C=0 TO 3
70  COLOR 5+C*3,5+C*2,C   ;第70行指定0～3四种颜色
80  NEXT                  ;屏幕上有4块被配出不同颜色
90  LOCATE 0,20
```

CGEN

功　能:确定背景面和卡通面所采用的符号图表。

格　式:CGEN　n

n→组合代码0～3

缩　写:CGE.

解　释:(1)表4.4表示了组合代码所表示的含义。表中的A是指符号图表A;B是指符号图表B。在附录中有符号图表的含义,参见附录。

符号图表A:各种卡通图案(封底)。

符号图表B:各种符号、数字及背景图案等(封三)。

(2)利用CGEN指令可把A中所示的卡通图案和B中所示数字、英文及背景图案,确定在卡通面还是在背景面上使用。可以把卡通图案表示在背景面,也可把符号表示在卡通面。一般,已经事先设定为CGEN2。

利用[CTRL]+[D]键可以设定CGEN2。即两者等价,但一方只需按键,另一方要输入5个符号。

表4.4　符号图形的设定表

n	背景面	卡通面	含义
0	A	A	背景面与卡通面均使用A。
1	A	B	背景面使用A,卡通面使用B。
2	B	A	背景面使用B,卡通面使用A。
3	B	B	背景面与卡通面均使用B。

程序例:

```
10  REM * CGEN *                     ;20行进行一系列初始化
20  CLS:SPRITE ON:CGSET 0,1          ;30～50行把符号表显示在屏幕上
30  FOR I=32 TO 255
40  PRINT CHR$(I);
50  NEXT                             ;60行定义"飞鱼"
60  DEF SPRITE 0,(0,1,0,0,0)=CHR
    $(64)+CHR$(65)+CHR$(66)+CHR$(67)
70  SPRITE 0,100,150                 ;70行设定飞鱼出现的位置
80  PAUSE 100:BEEP
90  CGEN 0                           ;90～150行设定符号表
100  PAUSE 100:BEEP                  ;四种变化循环出现
110  CGEN 1                          ;CHR$64,65,66,67对应于图A是飞鱼,
120  PAUSE 100:BEEP                  ;对应于图B是A、B、C,请参阅附录。
130  CGEN 3
140  PAUSE 100:BEEP
150  CGEN 2
160  GOTO 80
```

CLS

功　能:清除屏幕并使光标返回左上角。

格　式:CLS

缩　写:CL.

解　释:清除背景面同时变为黑色。但不影响卡通面的图形。在把 BG 图形复制到背景面中(利用 BG 绘图程序制作的图形)时,必须先使用 CLS 指令,再执行 VIEW 指令,否则出错!

CGSET

功　能:决定背景面和卡通面所使用板代码。

格　式:CGSET〔m〕〔,n〕

m→背景面的板代码 0～1,m 称为背景代码。

n→卡通面的板代码 0～2,n 称为卡通代码。

缩　写:CG.

解　释:(1)CGSET 指令是从预先准备好的色彩组合中挑选颜色,CGSET 只是指定板代码。色彩组合可参见封 3,图 4.4 是其示意

(2)背景面有两种组合(板代码)可以指定。卡通面有 3 种组合(板代码)可供指定。

(3)图中卡通板代码表示的三组图形中又分别有 0,1,2,3 四个代码。这四个代码称配色代码,每个配色代码指定三种色彩。配色代码不在 CGSET 指令中指定,而在第二章学过的两个指令:

DEF SPRITE……

DEF MOVE(N)=……

中指定。卡通代码 n 和配色代码一起可以指定卡通及其各种动作的颜色。

(4)图中背景板代码表示的两组图形也分别有四个配色代码,这四个配色代码在 COLOR 指令中指定。

(5)CGSET 指令的缺省值为:

CGSET 1,1

即当计算机进入 BASIC 状态后,卡通面和背景面均为代码 1 的状态。

(6)板代码是对整个屏幕面板的操作代码,当板代码一改变,卡通面上的所有卡通,或者背景面上的所有背景都将改变。如何改变由 CGSET 指令进行指定。

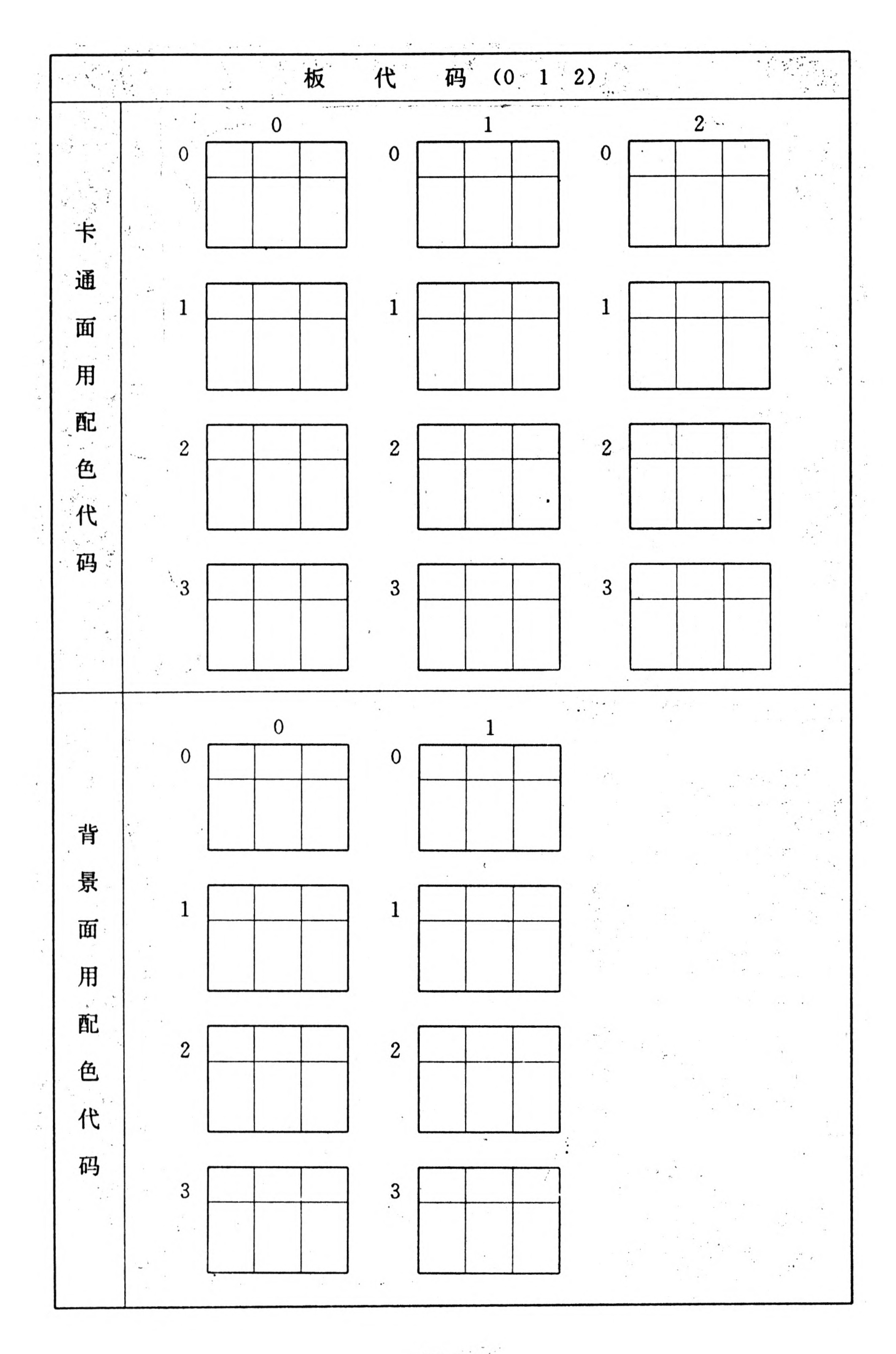


图 4.4

为了便于理解,对卡通面代码说明如下:

```
  CGSET 1,1                              ;指定卡通面板代码为 1
DEF SPRITE 0,(0,1,0,0,0)=……
DEF MOVE(N)=SPRITE(0,3,1,10,0,3)         ;指定配色代码分别为 0 和 3
```

为了便于理解,对背景面代码说明如下:

```
  CGSET 1,1                              ;指定背景面代码为 1
  COLOR 10,10,3                          ;座标(10,10)附近的四块为 3 号配色代码。
```

PALET

功　能:把配色代码(0~3)所指定的 3 种颜色代码重新设定。

格　式:PALET $\left\{\begin{matrix}B\\S\end{matrix}\right\}$ n,C1,C2,C3,C4

B→表示背景面

S→表示卡通面

n→配色代码(0~3)

C1→指定底背景颜色

C1~4→颜色代码

C2,C3,C4 分别对应配色代码所指定的左、中、右三块的配色。如:

配色代码 0			
	C2	C3	C4

颜色代码

解　释:(1)表 4.5 中的代码可以用以改变卡通、背景与底背景的颜色。

(2)底背景的颜色被显示在卡通面、背景面之后。

(3)颜色由彩色发生器产生。

(4)16 进制和十进制同时标在配色代码表示的图形中,请参见封 3。下面试举一例:

54 36	22 16	2 02

上行为 10 进制
下行为 16 进制

查表 4.5 可知,10 进制 54 对应 16 进制 36,22,对应于 16,2 对应于 02。

(5)下面试举一个颜色改变的例子。

```
CGSET 1,0
DEF SPRITE 0,(0,1,0,0,0)=……
```

在上例中首先指定卡通板代码为 0;再指定 0 号卡通板中的 0 号配色代码,其代码为图 4.5(a),当执行下面指令后颜色代码改就成图 4.5(b)。在程序中使用 PALET 指令,一般用在指定板代码和配色代码的指令之后。

52 36	22 16	2 02

(a)

48 30	25 19	18 12

(b)

图 4.5

表 4.5　颜色代码

	16 进制	10 进制	16 进制	10 进制	16 进制	10 进制	16 进制	10 进制	
	00	0	10	16	20	32	30	48	灰～白
兰	01	1	11	17	21	33	31	49	
	02	2	12	18	22	34	32	50	
	03	3	13	19	23	35	33	51	
	04	4	14	20	24	36	34	52	
红	05	5	15	21	25	37	35	53	过渡色
	06	6	16	22	26	38	36	54	
	07	7	17	23	27	39	37	55	
	08	8	18	24	28	40	38	56	
绿	09	9	19	25	29	41	39	57	
	0A	10	1A	26	2A	42	3A	58	
	0B	11	1B	27	2B	43	3B	59	
	0C	12	1C	28	2C	44	3C	60	
	0D	13	1D	29	2D	45			黑色
	0E	14	1E	30	2E	46			
	0F	15	1F	31	2F	47			

暗————————————————————————明

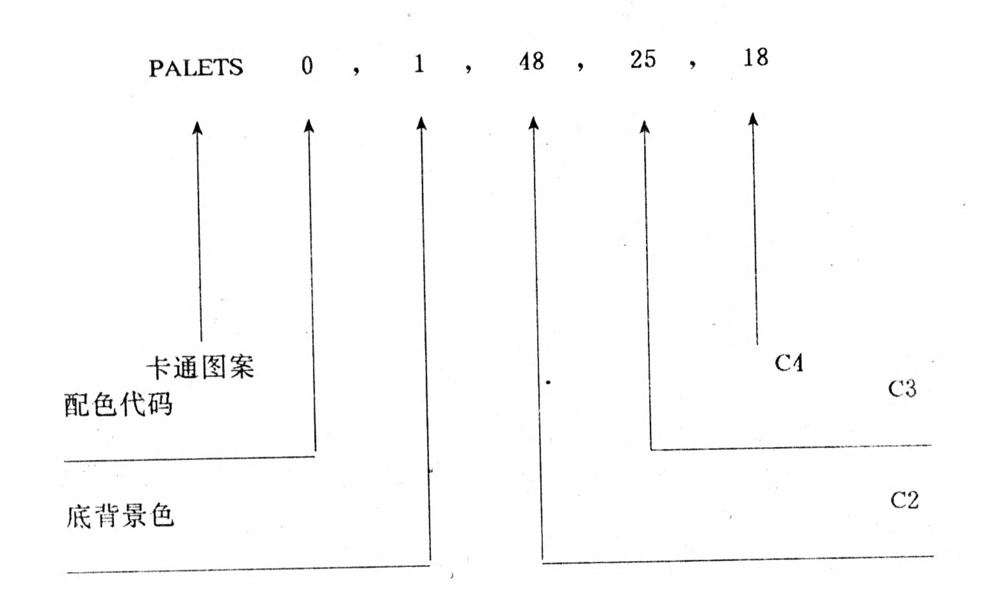


图 4.5

这里，先执行 CGSET 1,0 再执行 PALETS 指令则与图 4.5 的例子一样结果。见下例：

程序例：10　REM * PALET *

```
20  SPRITE ON
30  DEF SPRITE 0,(3,1,0,0,0)=CHR$(88)
    +CHR$(89)+CHR$(90)+CHR$(91)
40  SPRITE 0,100,150
    RUN
OK
```

CGSET 指令的缺省值为:CGSET1,1,所以卡通板代码为 1,配色代码为 3(第 30 行中定义)。

```
CGSET 1,0              ;设定卡通板代码 0。
OK
PALETS 0,&H00,&H30,&H19,&H12          ;用 16 进制表示颜色代码
OK
```

上面共讲了多种代码,为了便于学习特小结如下:

表 4.6 各种代码比较

	板代码	配色代码	颜色代码	底背景颜色代码
卡通面用	0 1 2	0 1 2 3	C_2 C_3 C_4	C_1
背景面用	0 1	0 1 2 3	C_2 C_3 C_4	C_1
所用指令	CGSET	DEF SPRITE DEF MOVE COLOR	PALET	PALET

4.7 MOVE 系列指令

MOVE 系列指令是控制卡通图案在卡通面上运动的指令群。MOVE 指令的运用可使游戏程序的编写更加简单,动作更加生动。

DEF MOVE

功　能:定义卡通动作。

　　(可使用的卡通共有 16 种)

格　式:DEF MOVE(n)=SPRITE(A,B,C,D,E,F)

　　n→被定义的卡通动作(0~7)

　　A→卡通种类见封底(0~15)

　　B→卡通的运动方向(0~8)

　　C→运动速度(1~255)

　　D→位移量(1~255)

　　E→优先度(0~1)

　　F→配色代码

缩　写:DE. M.

解　释：(1)DEF MOVE(n)指令最多可以定义卡通在8个方向上的动作。为了产生动画效果，每一个动作至少应由两个卡通姿势组成，当由一个姿势组成的动作，比如玛丽沃跳组成动作，就不会产生动画效果。卡通姿势较少，在不同运动方向上的动作容易重复。比如：乌龟在8个方向上都只能是“走”一个动作。

(2)当指定了卡通运动方向，卡通的动作由DEF MOVE(n)自动给出。不同方向给出不同的动作。

(3)用DEF SPRITE可以直接调出封底图A中的卡通姿势，而用DEF MOVE(n)指定的是卡通动作。

(4)卡通图案一共有16种，参见封底图A。卡通图案有时也被称为卡通姿势。不经定义的卡通不可能在程序中使用。16种卡通(图案)是：

0:玛丽沃	4:企　鹅	8:太空杀手卫星	12:激　光
1:丽　莎	5:火　球	9:太　空　船	13:乌　龟
2:苍　蝇	6:车	10:爆　炸	14:螃　蟹
3:飞　鱼	7:太空站	11:妖　怪	15:鸟

(5)卡通运动方向一共有8个如图4.6所示。

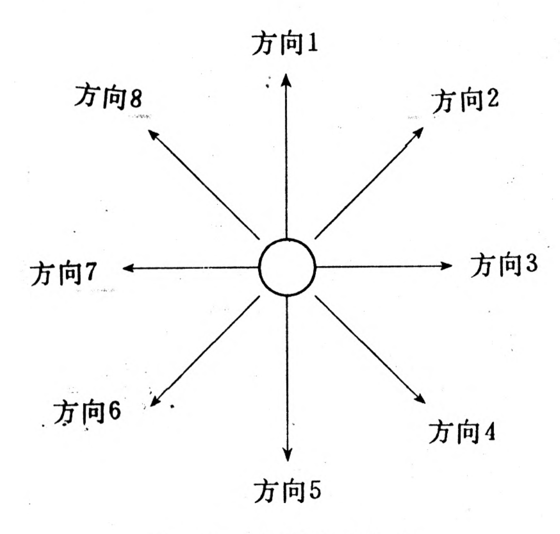


图4.6　卡通运动方向

卡通运动方向按顺时钟排列，向上为方向1静止为0。

(6)运动速度是指卡通在屏幕上的运动速度。最快一秒钟可运动60个点(指定C=1)，当指定C=255时，则255秒运动60个点。由此类推当指定C=20时，则20秒运动60个点。当屏幕接点计算大小时是256个点宽，240点长。

(7)位移量是指卡通运动距离，整个移动范围是2×255个点。由于每次移动两个点，当指定D=255时，卡通可以在屏幕出现 两次。

(8)优先度表示卡通是在卡通0面还是卡通1面上显示。卡通0面在背景面之前，卡通1面在背景面后。

(9)配色代码是指在卡通面上所使用的代码。要指定配色代码之前先用CGSET指令指定板代码，当用不同的板代码时，即使配色代码相同，含义也完全不同，请参看表4.6和封3

中的内容。

MOVE

功　能：使卡通开始运动。

格　式：MOVE n_0〔，n_1，n_2，n_3，n_4，n_5，n_6，n_7〕

n_0～n_7 是与 DEF MOVE 指定的卡通动作代码(0～7)一致。

缩　写：M.

解　释：(1)首先在执行了 SPRITE ON 之后，用 MOVE 指令可使在 DEF MOVE 指令中所定义的卡通开始运动。

(2)一执行 MOVE 指令，如果在 DEF MOVE 指令所指定的运动没有结束，即使程序执行完了，卡通也会按 DEF MOVE 指令中所定义的位移量、方向、速度把动作做完。

(3)在同一水平线上可以显示出 4 个卡通图案。

CUT

功　能：用 MOVE 指令可使卡通运动，而 CUT 指令则可令运动中的卡通停止。

格　式：CUT n_0〔，n_1，n_2，n_3，n_4，n_5，n_6，n_7〕

缩　写：CU.

解　释：(1)停止卡通运动，执行下一个 MOVE 指令，卡通可从被停止的位置继续按在 DEF MOVE 所定义的值运动。

(2)CUT 指令可同时定义 8 个卡通停止。

(3)停止的卡通不会消失。

程序例：

```
CUT 0,1,2                 ;直接键入指令执行
OK
CUT 0,1,2,3,4,5,6,7
```

ERA

功　能：停止卡通运动并让被停止的卡通消失。

格　式：ERA n_0〔，n_1，n_2，n_3，n_4，n_5，n_6，n_7〕

缩　写：ER.

解　释：(1)用 MOVE 指令可让在 DEF MOVE 所定义的卡通运动，用 CUT 可让卡通停止。而用 ERA 可让运动或停止的卡通消失。

(2)可同时指定 8 个卡通。

(3)消失的卡通可用 MOVE 指令再次起动。卡通将从消失的位置出现，并按 DEF MOVE 指令中的规定运动。

(4)再执行 ERA 之前要先用过 MOVE 指令。

程序例：

```
10  REM * MOVE *
20  CLS
30  SPRITE ON
40  DEF MOVE(0)=SPRITE(0,3,4,255,0,2)
50  MOVE 0
RUN                      ;玛丽沃开始行走
ERA 0                    ;玛丽沃消失
```

```
MOVE 0                 ;玛丽沃继续行走
```

POSITION

功　能:在用 MOVE 指令让卡通运动之前,给出卡通运动的初始座标。

格　式:POSITION N,X,Y

N→所定义卡通动作代码(0—7)

X→水平方向座标

Y→垂直方向座标

(在卡通面上 X、Y 的有效范围是:X:0～240;Y:5～220)

缩　写:POS.

解　释:(1)给出卡通运动的初始座标,不指定时,缺省值为 X=120,Y=120.

程序例:

```
10  REM * POSITION *
20  CLS:SPRITE ON
30  DEF MOVE(0)=SPRITE(11,3,2,10,1,2)  ;定义卡通向右移动 20 个点(2×10)
40  X=RND(256):Y=RND(240)              ;使用随机函数产生任意的 X,Y
50  PRINT"X;Y=";X;",";Y                ;打印出 X,Y 的值
60  POSITION 0,X,Y                     ;指定卡通运动初始位置
70  MOVE 0                             ;使卡通开始运动
80  PAUSE 80                           ;暂停
90  GOTO 20                            ;返回到 20 行
```

XPOS

功　能:求出在 DEF MOVE 所定义的卡通水平位置座标。

格　式:XPOS(n)

n→DEF MOVE 中所定义卡通动作代码(0～7)

缩　写:XP.

解　释:XPOS(n)指令可以测出 n 卡通动作的 X 座标值。当动作方向发生改变时,利用 POSITION 指令 和 XPOS 指令使动作连贯起来。

程序例:参见程序 3.7

YPOS

功　能:求出在 DEF MOVE 所定义卡通动作的垂直位置座标。

格　式:YPOS(n)

n→DEF MOVE 中所定义卡通动作代码(0～7)。

缩　写:YP.

解　释:YPOS(n)可以测出 n 卡通动作的 Y 座标值。利用 POSITION 指令和 YPOS 指令可使卡通动作连贯起来。

程序例:

参见程序 3.7

MOVE(n)

功　能:用于测试 DEF MOVE 所定义的 n 卡通动作是否结束。

格　式:MOVE(n)

n→DEF MOVE 所定义的卡通动作代码。

缩　写:M.(n)

解　释:当 n 卡通开始运动后动作未结束时,MOVE(n)=0;当动作结束后 MOVE(n)=−1。这里 n 为 0～7。

程序例:

```
10  REM * MOVEN *
20  SPRITE ON:CGSET 1,0
30  DEF MOVE(0)=SPRITE(0,3,1,150,0,0);定义 0 号卡通动作向右移动 300 个点
40  MOVE 0                               ;让 0 号动作开始运动
50  IF MOVE(0)=-1 THEN PRINT"MOVE(0)=";MOVE(0):GOTO 50;判定动作是否结束
60  PRINT"MOVE(0)=";MOVE(0)                ;结束时打印 MOVE(0)
70  END
```

4.8　特殊功能指令

KEY

功　能:用于设定功能键。

格　式:KEY N,"字符串"

N→所设定功能键的号码(1～8)

字符串→在功能键中所设定的内容

缩　写:K.

解　释:在键盘的左侧有 8 个功能键[F1]—[F8],这些功能键里的内容可以由用户自己设定。在出厂时功能键被预先设置了下述功能。用户可自行改变。

F1:LOAD(M)

F2:PRINT

F3:GOTO

F4:CHR $ (

F5:SPRITE

F6:CONT(M)

F7:LIST(M)

F8:RUN(M)

带有(M)的指令表示不必按回车键直接按该功能键即可执行该指令。

(M)可以被设定。设定方法如下:

KEY N,"字符串"+CHR $ (13)

程序例: KEYLIST　　　　;KEYLIST 指令可以用于看功能键的内容

```
F1  LOAD(M)
F2  PRINT
F3  GOTO
F4  CHR$(
F5  SPRITE                          ;F1 — F8 功能键的内容
F6  CONT(M)
F7  LIST(M)
F8  RUN(M)
OK
KEY1,"SVAE"+CHR$(13)              ;设置 F1 的内容
OK
KEY5,"DEF MOVE("                  ;设置 F5 的内容
OK
KEYLIST                           ;用 KEYLIST 列表,可以看出 F1 和 F5 功能键的内容
                                   被改变了。
OK
F1  SAVE(M)
F2  PRINT
F3  GOTO
F4  CHR$(
F5  DEF MOVE(
F6  CONT(M)
F7  LIST(M)
F8  RUN(M)
OK
```

KEYLIST

功　能:把功能键的内容列表。

格　式:KEYLIST

缩　写:K.L.

解　释:一执行 KEYLIST 指令,功能键的内容被显示在屏幕上。

PAUSE

功　能:暂时停止程序执行。

格　式:PAUSE[n]

　　n→0—32767

缩　写:PA.

解　释:一执行 PAUSE 指令便按 n 所规定的时间暂停。若省去 n,程序将一直暂停,按下任意一个键,程序才继续执行。

程序例:
```
10  REM * PAUSE *
20  FOR I=0 TO 10
```

```
30  BEEP:PAUSE 10
40  NEXT
50  PAUSE               ;程序在第 50 行暂停,只有按下任意一个键程序才能继续执行
60  FOR I=0 TO 10
70  PLAY"C":PAUSE 20
80  NEXT
```

SYSTEM

功　能:返回 FBASIC 菜单屏幕

格　式:SYSTEM

缩　写:S.

解　释:当由 FBASIC 菜单屏幕通过选择数字[1]进入 BASIC 状态后,用 SYSTEM 指令可以返回 FBASIC 菜单屏幕。

VIEW

功　能:调出用 BG 绘图程序所写的背景

格　式:VIEW

缩　写:V.

解　释:(1)在 BASIC 状态用 VIEW 指令可把用 BG 绘图程序所写的背景画面调到屏幕上。这时执行 BASIC 程序不会影响背景画面。

(2)VIEW 指令一般用在程序的最前面。

4.9 声音控制指令

BEEP

功　能:使喇叭发出"嘀"的一声。

格　式:BEEP

缩　写:B.

PLAY

功　能:演奏音乐

格　式:PLAY"音乐字符"

缩　写:PL.

解　释:根据"音乐字符"所指定的乐曲进行演奏。"音乐字符"是指:音符、8 度音、音量、和音、主音和音色。分别说明如下:

(1)拍节→用 T1~T8 指定拍节

T1(快)↔T8(慢)

(2)音色→用 Y0—Y3 指定音色

Y0:12.5%　　　;音色在指定音量后指定

Y1:25%　　　;

Y2:50%

Y3:75%

(3)主音

M0→用 V0~V15 指定音量

0(小)←音量→15(大)

M0V15 表示音量为 15

M1→用 V0～V15 指定主音长度

M1V3 表示音长为 3

8 度音→用 O0～O5 指定 8 度音

O0(低音)←→O5(高音)

(4)音符:用下表符号 C～B 表示:

表 4.7 音符的表示方法

音 符	指定方法
1	C
1#	#C
2	D
2#	#D
3	E
3#	#E
4	F
4#	#F
5	G
5#	#G
6	A
6#	#A
7	B

(5)休止符→用 R0～R9 指定休止长度。

(6)长度→音符、休止符后附有 0～9 的整数来指定声音和休止符的长度。表 4.8 表示了声音长度和数字的对应关系。

4 分音符相对于数字 5,可以写成 C5R1,当不指定数字时,后面的音长与前面一致。

(7)和音→演奏 2 重或 3 重和音时需加冒号。比如:

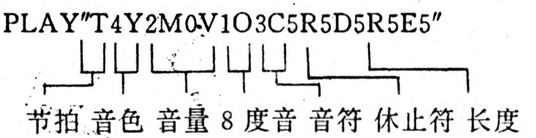
表 4.8 声音长度与数字的对应关系

声 音 长 度		对 应 的 数 字
$\frac{1}{8}$(32 分音符 𝅘𝅥𝅰)	𝅀	0
$\frac{1}{4}$(16 分音符 𝅘𝅥𝅯)	𝄿	1
$\frac{3}{8}$(付 16 分音符 𝅘𝅥𝅯.)	𝄿.	2
$\frac{1}{2}$(8 分音符 ♪)	𝄾	3
$\frac{3}{4}$(付 8 分音符 ♪.)	𝄾.	4
1(4 分音符 ♩)	𝄽	5
$1\frac{1}{2}$(付 4 分音符 ♩.)	𝄽.	6
2(2 分音符 𝅗𝅥)	𝄼	7
3((付 2 分音符 𝅗𝅥.)	𝄼.	8
4(全音符)	𝄻	9

"系统 1:系统 2:系统 3:"各系统须指定音符,声音长度、拍节和音色等。

(8)缺省值→T4、M0、V15、O3、C5、R5 一经指定参数后，若不再指定，各参数不变。

试举一例说明各参数的含义：



详细的音乐设计法另书说明。

程序例：

参见关于 RND 指令的说明。

下面语句可直接键入：

详细的音乐设计法另书说明。

程序例：

参见关于 RND 指令的说明。

下面语句可直接键入：

参见关于 RND 指令的说明。

下面语句可直接键入：

4.10 函数

函数分为数值函数、字符函数和特殊函数。下面介绍如下：

4.10.1 数值函数

ABS(X)

功　能：把数值转换成绝对值。

格　式：ABS(X)

缩　写：AB.

解　释：X 的取值范围是－32768～＋32767。经过绝对值指令处理过的数值均为正数。

程序例：

```
PRINT ABS(41)
41
OK
PRINT ABS(-41)
41
OK
PRINT ABS(10—34)
24
OK
```

SGN

功　能：用来测试变量是正、是负还是零。

格　式：SGN(X)

缩　写：SG.

解　释：利用这个符号函数，可以测试变量变化的情况。

X＞0　　SGN(X)＝1

X＝0　　SGN(X)＝0

X<0　　SGN(X)=-J

程序例：

```
PRINT SGN(41)
1
OK
PRINT SGN(0)
0
OK
PRINT SGN(-41)
-1
OK
```

RND

功　能：这个函数可让计算机随机(任意)地产生一些数据。

格　式：RND(X)

　　X→在(1～32767)范围内选择

解　释：RND 指令被称为随机函数，意思是如果给出 X=100，RND(X)就可在 0～32767 的范围内任意产生一个数。通常用来设计游戏中的不固定目标，如敌机、炸弹等等。

程序例：

```
10  REM * RND *
20  PLAY"M1V6T2Y1"
30  A$="CDEFGABR"              ;设定音符 7 个加一个休止符。
40  N=RND(8)
50  B$=MID$(A$,N+1,1)          ;从 A$ 中取出一个，取出哪一个由第 40 行定。MID
                               $ 指令见字符函数中 MID$ 指令的解释。
60  PRINT B$;
70  PLAY B$                    ;为演奏
80  GOTO 40
```

利用这个程序可以奏出不固定乐曲的音乐。

4.10.2 字符函数

ASC

功　能：把字符转换成计算机内部使用的代码，即 ASCII 码。

格　式：ASC(字符串)

缩　写：AS.

解　释：ASCII 代码是国标上常用的一种计算机内代码，它是数字、字母及常用符号的代码。对于研究计算 机内部的人员，机内码会经常用到。对于学习 BASIC 语言的人偶而也会碰到。附录中符号码 B 中代码便是机内代码。比如下面便是字符与内码的对应表。

表 4.9 字符内码的对应

字　符	内　　码	
	10 进制	16 进制
A	65	41
B	66	42
C	67	43
D	68	44
·	·	·
·	·	·
·	·	·

程序例：

```
10  REM * ASC *
20  INPUT A$                  ;输入一个字符
30  A=ASC(A$)                 ;把输入字符转成机内码
40  PRINT A$;"………";A          ;把字符和内码打印出来
50  GOTO 20
```

CHR$

功　能：把 256 个机内码转换成所代表的字符(图形)。

格　式：CHR$(X)

缩　写：CH.

解　释：CHR$的功能与 ASC 正好相反。X：0～255。

程序例：

```
10  REM * CHR$ *
20  INPUT"0—255";A            ;输入一个数字
30  A$=CHR$(A)                ;把数字转成字符
40  PRINT A;"………";A$          ;打印出来
50  GOTO 20
```

VAL

功　能：把字符型数据转换成数值型数据。

格　式：VAL(字符变量)

解　释：(1)如果字符串变量开头没有＋、－、&、空格或数字，则 VAL(字符变量)为零。

(2)可把 16 进制的数转成 10 进制。

(3)数字不超过－32768～＋32767。

程序例：

```
10  REM * VAL *
20  INPUT A$                  ;输入字符型数据
30  A=VAL(A$)                 ;把字符型数据变成数值型
40  PRINT A$,A                ;打印结果
```

```
50  GOTO 20
RUN
? &A                              ;输入 16 进制数 A
&A        10                      ;打印结果,十进制数为 10
? -&A                             ;输入-A,& 为十六制数代号也可写成"&H"
-&A       -10                     ;打印结果
```

〈例 2〉

```
10  REM * VAL *
20  INPUT A$
30  A=VAL("&H"+A$)
40  PRINT A$;"………";A
50  GOTO 20
```

STR $

功　能:把数值型数据转成字符型数据。

格　式:STR $ (X)

解　释:有时需要把数值型数据转成字符型数据以便参加运算。

程序例:

```
10  REM * STR$  *
20  INPUT A:INPUT B
30  A$=STR$(A):B$=STR$(B)
40  PRINT A$+B$;"………";A+B
50  GOTO 20
```

运行上述程序可以看出,A \$ +B \$ 与 A+B 的值是不一样的。尽管从表面上看 A \$ 等于 A,B \$ 等于 B。但是两者数据类型完全不一样。前者是字符型后是数值型。字符型数据求不出代数和。

HEX $

功　能:把 10 进制数转变成 16 进制数。

格　式:HEX $ (X)

X→在-32768~+32767 范围内取值。

程序例:

```
10  REM * HEX$ *
20  FOR I=0 TO 20
30  PRINT"&H";HEX$(I);"=";I
40  NEXT
```

这个程序把 0~20 的 10 进制数与 16 进制数进行了比较。

LEFT $

功　能:从字符串左侧取出字符。

格　式:LEFT $ (字符串,n)

字符串→最多 31 个字符。

n→表示从左边取出的字符数

解　释:LEFT $ 的取字符方向与 RIGHT $ 相反。

程序例:

```
10  REM * LEFT $ *
20  A $ ="ABCDE"
30  FOR I=1 TO 5
40  PRINT LEFT $ (A $ ,I)            ;从左侧取字符
50  NEXT
RUN                                  ;下面是运行结果
A
AB
ABC
ABCD
ABCDE
OK
```

把 40 行换成:

```
40 PRINT LEFT $ ("ABCDE",I)
```

并删掉 20 行,即:

```
20↓
```

运行后可得同样结果。

RIGHT $

功　能:从字符串右侧取出字符。

格　式:RIGHT $ (字符串,n)

字符串→最多 31 个字符。

n→表示取出的字符数。

缩　写:RI.

解　释:用法与 LEFT $ 相对

程序例:

```
10  REM * RIGHT $ *
20  A $ ="ABCDE"
30  FOR I=1 TO 5
40  PRINT RIGHT $ (A $ ,I)
50  NEXT
RUN
E
DE
CDE
BCDE
ABCDE
OK
```

MID $

功　能:取出字符串中任意字符和取出多个字符。

格　式:MID $(字符串,开始位置,n)

字 符 串→最多 31 个

开始位置→从左侧开始计算从第几个字符开始取。

n→取出的字符数。

缩　写:MI.

解　释:MID $ 指令可以从中间、左边、右边任意取字符。

程序例:

```
10  REM * MID $ *
20  A $ ="ABCD"
30  FOR I=1 TO 4
40  PRINT MID $(A $ ,I,1)
50  NEXT
RUN                    ;下面是执行结果
A
B
C
D
OK
```

LEN

功　能:用于求字符串的长度。

格　式:LEN(字符串)

缩　写:LE.

解　释:在计算字符串长度时,空格和非印刷字符也计算在内,长度为 0～31。

程序例:

```
10  REM * LEN *
20  FOR I=1 TO 28
30  A $ ="A"+A $
40  L=LEN(A $ )
50  PRINT A $
60  NEXT
70  PRINT L
```

INKEY $

功　能:检测键盘输入。

格　式:INKEY $ $\left\{\begin{matrix}(n)\\ \text{省略}\end{matrix}\right\}$

缩　写:INK

解　释:(1)从键盘读入一个字符。当 n=0,程序在有 INKEY $ 指令的行停止,直到输入了一个字符后才接着运行程序,n 常取零。

(2)当 n 被忽略,程序不会停。当有键按下时,该字符便成为 INKEY $ 的值。

程序例:〈1〉

```
10  REM * INKEY$(0) *                  ;
20  A$=INKEY $(0)                      ;没有键按下程序停在这一行
30  IF A$="A"THNE PRINT A$             ;
40  BEEP:GOTO 20                       ;若所按的不是A,发出嘀的声跳回 20 行
```

〈2〉

```
10  REM * INKEY $ *                    ;
20  A$=INKEY $                         ;没有键按下程序在这一行不停
30  IF A$="A"THNE PRINT A$             ;
40  BEEP:GOTO 20                       ;没有键按下或所按的不是A,将不断发出嘀声
```

4.10.3 特殊函数

POS

功　能:给出光标当前的列数,即水平方向的变化。

格　式:POS(0)

解　释:水平方向的变化即X方向的变化,左边的X座标为0,右边为25。

程序例:

```
10  REM * POS *
20  CLS
30  FOR X=0 TO 25
40  LOCATE X,0
50  PRINT POS(0)
60  PAUSE 10
70  NEXT
```

FRE

功　能:看内存剩余空间。

格　式:FRE

缩　写:FR.

解　释:利用FRE指令可以了解内存空间的剩余。

程序例:PRINT FRE

```
4030          ;这时内存还有 4030 个单元可以使用。
```

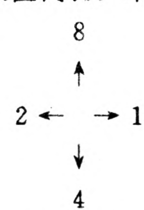
STICK

功　能:用来检测操纵器十字形方向键的上下左右。

格　式:STICK(X)　　X→取 0、1,表示是 1 号还是 2 号操纵器。

缩　写:STI

解　释:十字形方向键的不同方向用不同数值代表。即



上下左右各有一个数值,按的方向不同,STICK 函数的值也不同。

```
程序例:10  REM * STICK *
       20  S=STICK (0)
       30  IF S=0 THEN PRINT"S=0"
       40  IF S=1 THEN PRINT"S=1"
       50  IF S=2 THEN PRINT"S=2"
       60  IF S=4 THEN PRINT"S=4"
       70  IF S=8 THEN PRINT"S=8"
       80  GOTO 20
```

只要用 1 号操纵器便可以不断打印出字符。

从程序上还可以看出,假如有两个键同时按下时,将不打印结果,这时屏幕内容就不再滚动。

STRIG

功　能:用于测试操纵器 4 个按钮是否被按下。

格　式:STRIG(X)　　X→取 0,或 1,表示是 1 号还是 2 号操纵器。

缩　写:STR.

解　释:操纵器 4 个按钮的对应的数值为

1 号	选择 CELECT ▬	开始 START ▬	B A ● ●
	2	1	4 8
2 号			B A
			4 8

这个数便是 STRIG(X)函数的值。

```
程序例:10  REM * STRIG *
       20  T=STRIG(0)
       30  IF=1 THEN PRINT"START"
       40  IF=2 THEN PRINT"SELECT"
       50  IF=4 THEN PRINT"B"
       60  IF=8 THEN PRINT"A"
       70  GOTO 20
```

CSRLIN

功　能:给出光标的所在行,即

格　式:CSRLIN

缩　写:CSR.

解　释:光标垂直变化用 CSRLN 测试,水平变化由 POS(0)测试,两者合起来可以定出平面座标。

```
程序例:10  REM * CSRLIN *
       20  CLS
       30  FOR I=0 TO 20
       40  LOCATE I,I
       50  PRINT POS(0);",";CSRLIN
       60  PAUSE 20
```

```
70  NEXT
```

上述程序将在屏幕上沿 45° 角显示数字。

SCR $

功　能：求出背景面中所显示的字符和图形。

格　式：SCR $ (X,Y,Sw)

X→横座标(0～27)

Y→纵座标(0～23)

Sw→求出配色代码(0～1),0 可以省略。

缩　写：SC.

解　释：指定背景画面上的行和列,可以了解指定位置的字符或图形是什么。Sw 为 1 时可以求出图形的配色代码(0～3)。

程序例：

```
10  CLS
20  LOCATE 0,10                    ;第 20 行指定屏幕位置
30  PRINT"FAMILY－COMPUTER"        ;
40  PRINT"………"                    在屏幕上打印一些字符,
50  LOCATE 10,15                   ;指定位置
60  PRINT SCR $ (0,10)             ;
70  A $ =SCR $ (1,10)              ;
80  PRINT A $                      ;
90  C $ =SCR $ (1,10,1)            ;
100 PRINT"COLOR=";ASC(C $ )        ;60～100 行取出字符
110 END                            ;
```

4.11　控制卡通面指令

控制卡通面指令一共有 4 个,是 FBASIC 的特点之一,下面分别叙述。

DEF SPRITE

功　能：定义在卡通面显示的卡通姿势。

格　式：DEF SPRITE A,(B,C,D,E,F)＝字符串。

A→卡通姿势代码 0～7

B→配色代码 0～3(见封 3 图 B)

C→图形大小 0～1(0 表示图形为 8×8 点阵组成,即一个字符大小;1 表示图形为 16×16 点阵组成,即 4 个字符大小。)

D→优先度 0～1(0 表示利用卡通 0 面,即在背景面之前;1 表示利用卡通 1 面,即在背景面之后。)

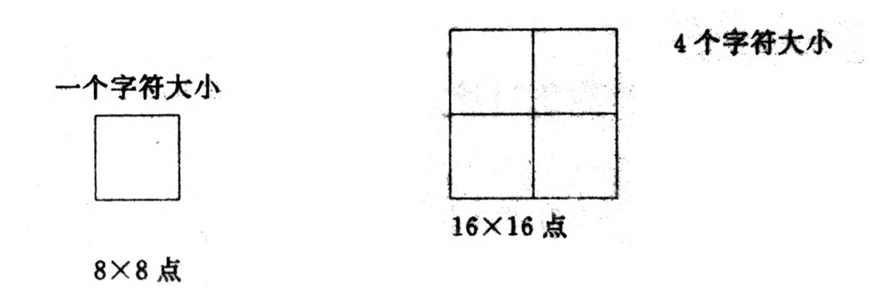
E→X 方向反转 0～1(0 表示与封底图 A 图形相同,1 表示图形发生左右反转。)

F→Y 方向反转 0～1(0 表示与封底图 A 图形相同,1 表示图形发生上下反转。)

缩　写：DE. SP.

解　释：在指定图形大小时可以指定两种大小(见下页)。关于卡通面的叙述参见 3.1 节。

图 4.7(a)和(b)是在 X 方向的反转图形。



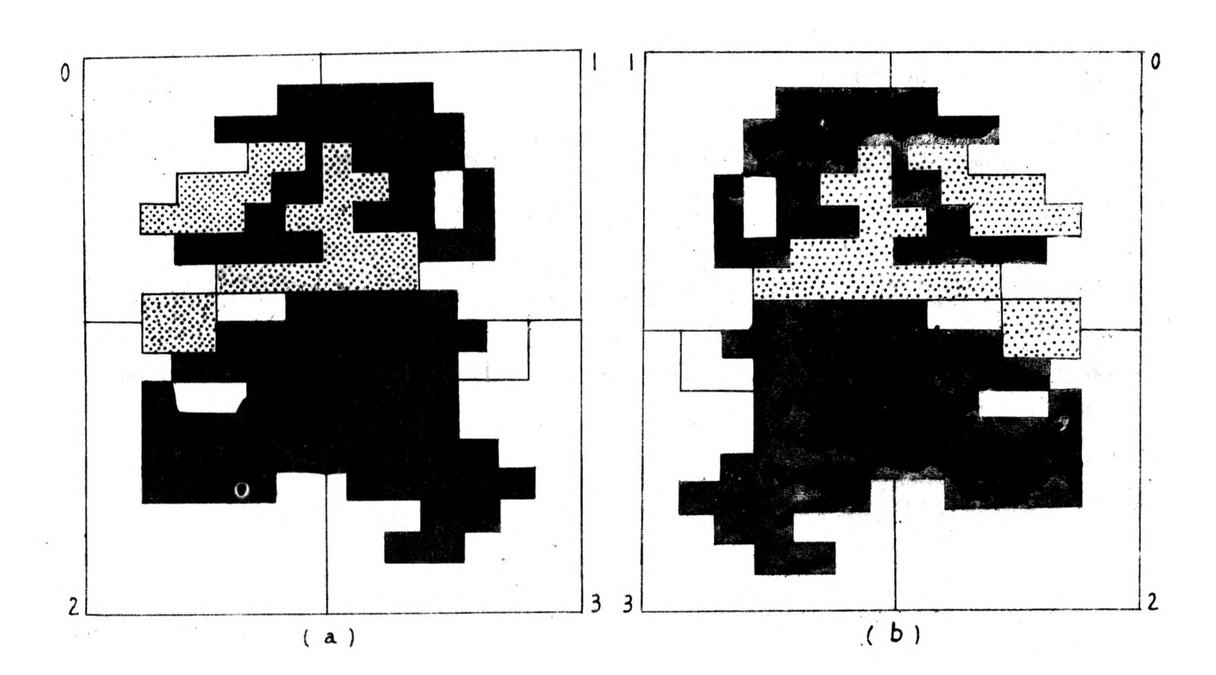


图 4.7

用 CGEN 指令进行设定，可以把符号、字母用在 DEF SPRITE 指令，其中字符串指令 CHR $ (N) 中的 N，可以使用符号图表 B 中的代码。

程序例：
```
10  REM * DEF SPRITE *
20  SPRITE ON
30  DEF SPRITE 0,(0,1,0,0,0)=CHR $ (0)+CHR $ (1)+CHR $ (2)+CHR $ (3)
40  SPRITE 0,100,130
```

程序 CHR $ (N)中的 0,1,2,3 分别对应图 4.7(a)4 角所标的号码，也对应符号图表 A 中的玛丽沃的代码。

SPRITE

功　能：将由 DEF SPRITE 所定义的卡通姿势在屏幕任意位置上显示或消除。

格　式：SPRITEn[,X,Y]

n→卡通姿势代码 0～7。

X→卡通面横座标 0～240。

Y→卡通面纵座标 5～220

缩　写:SP.

解　释:(1)当省略X,Y座标,则会在屏幕上清除n号卡通。指令格式为:

SPRITE n

n→卡通姿势代码0～7

(2)在同一座标上显示多个卡通时(发生卡通重叠时),代码小的能被看到(在最前面)。

(3)在同一水平只能显示4个卡通。

(4)背景面和卡通面座标不同,座标之间的关系为:

$x=(X\times8)+16$

$y=(Y\times8)+24$

其中:x,y是卡通面座标。

X,Y是背景面座标。

对于背景面和卡通面的详细讨论请参见3.1节、3.2节和3.11节。

SPRITE ON

功　能:打开卡通面。

格　式:SPRITE ON

缩　写:SP.O.

解　释:在执行DEF SPRITE或DEF MOVE指令之前,卡通面必须先打开。卡通面一旦被打开只要不执行SPRITE OFF就将一直打开。

SPRITE OFF

功　能:关闭卡通面。

格　式:SPRITE OFF

缩　写:SP.OF.

解　释:(1)关闭卡通面,所有卡通将消失。

(2)使卡通面无法重叠在背景面上。

·MOVE系列指令与SPRITE系列指令比较

控制卡通运动的指令是FBASIC的特点,控制卡通运动的指令一共有两套,即MOVE系列与SPRITE系列。从使用的方便程度上,MOVE系列要方便一些。因为MOVE系列是用于控制"动作"的,而SPRITE系列是用于控制"姿势"的"

在控制卡通方向上,SPRITE系列也是无能为力而MOVE系列可以定义8个方向的运动。

五、程序实例及说明

下述的 BASIC 程序充分发挥了 FBASIC 的特长，通过下面的程序例子，读者会对每个指令的含义和它们在程序中的作用有更深的体会。

5.1 学习程序实例的注意事项

对于本章给出的游戏例题，在输入和修改时应该注意下述几点：

▲输入程序时的注意事项

(1) 输入程序时，应正确地输入行号和语句，并符合语法规则。如下例：

```
 5  CLS
10  SPRITE ON
20  CGSET 1,1
30  FOR N=0 TO 7
40  DEF MOVE(N)=SPRITE(0,N+1,3,255,0,0)
50  NEXT
60  MOVE 0,1,2,3,4,5,6,7
70  GOTO 60
80  END
```

当每一行语句较长，屏幕的一行显示不下时，继续输入时自动换行显示。

(2) 输入时应按程序清单输入，在 GOTO,GOSUB 或 IF—THEN 后的行号可以不加空格，如：GOTO0120

(3) 有时会发生内存不够用的情况，这时出现内存溢出错误。

▲程序修改或变更时的注意事项

(1) BASIC 程序的编制，变更或修改必须在消除 BG GRAPHIC(背景)画面后进行。如果保留背景画面进行程序变更或修改，会产生错误的结果。

(2) 要消去背景面，只需按 HOME 键，光标便回到左上角。

(3) 用 LIST 指令调出程序，进行修改或更正。

总之，编制并运行游戏程序的整个过程是：首先用 BG GRAPHIC 指令把背景面设计好，其次消去背景面，用 BASIC 编制控制卡通运动的程序并调试好。最后在程序开始时，用 VIEW 指令调出背景。这样，一个完整的游戏程序就完成了。

▲背景面的图案

书后封三图表 B 中的图案，即为设计背景用的图案，如图 5.1 所示。

它们共分为 13 组(从 A—M)，每组有 8 个图案(0—7)，全部存于机器的内存当中，用时调出即可。

BG 绘图程序在横 28 个字×纵 21 行的范围内可绘制背景。

下面是一个背景设计的例子

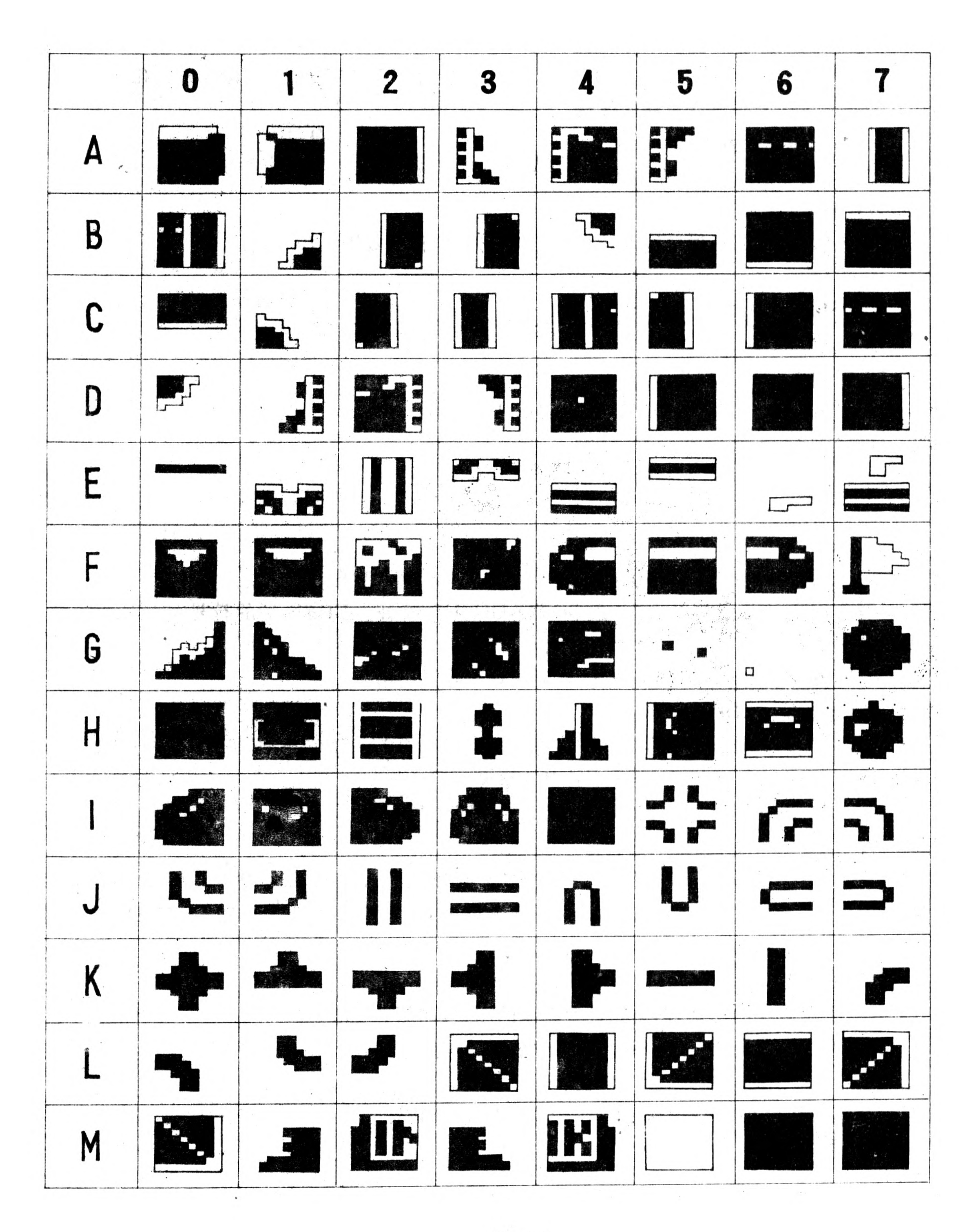


图 5.1

背景

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0										K60			J20			J20			K60									
1										K60			J20			J20			K60									
2										K60			J20			J20			K60									
3										K60			J20			J20			K60									
4										K60			J20			J20			K60									
5										J00	K50	K50	J20			J20	K50	K50	K10									
6													J20			J20												
7													J20			J20												
8	K50	K50	K50	K50	K50	K50	K50	170					J20			J20				160	K50	K50	K50	K50	K50	K50	K50	K50
9								K60					J20			J20				K60								
10								K60				160	J30	J30	J30	J30	170			K60								
11	J30	J30	J30	J30	J30	J30	J30	J30	J30	J300	J30	J20	G72		G72	G72	J20	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30
12												J20	G72				J20											
13												J20				G72	J20											
14	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J20	G72	G72		G72	J20	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30
15								K60				J00	J30	J30	J30	J30	J10			K60								
16								K60					J20			J20				K60								
17	K50	K50	K50	K50	K50	K50	K50	J10					J20			J20				J00	K50	K50	K50	K50	K50	K50	K50	K50
18													J20			J20												
19										160	K50	K50	J20			J20	K50	K50	170									
20										K60			J20			J20			K60									

其中：K50 意味着使用图表 B 中 K 组的第 5 号图形，配色代码为 0。

B73 意味着使用图表 B 组的第 7 号图形，配色代码为 3。

上述 K50、B73 等图表对应于封 3 图表 B 中的图形，可在 SELECT 状态(BG 程序中)下，输入与 K50、B73 对应的图形。在背景设计中，BG 绘图程序的 CHAR 状态下，可以直接输入字符

上面的背景设计对应着下面的图 5.2。

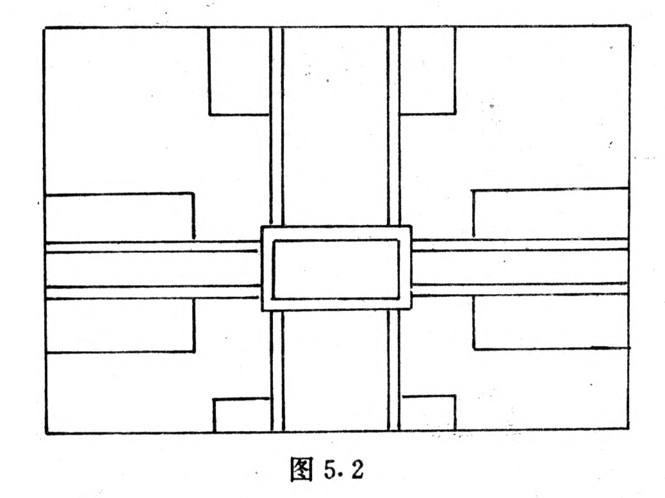


图 5.2

5.2 程序实例

程序例 1：跳马

程序清单

```
10 VIEW:CGEN 3:CGSET 1,1
20 DEF SPRITE 0,(0,0,0,0,0)=CHR$(&HC7)
30 PALETS 0,13,&H16,&H27,2
40 DEF SPRITE 2,(0,0,0,0,0)=CHR$(&HD7)
50 DEF SPRITE 1,(0,0,0,0,0)=CHR$(&HC7)
60 PALETS 1,13,&H16,&H17,4
70 DEF SPRITE 3,(1,0,0,0,0)=CHR$(&HD7)
80 SPRITE ON
```

```
90  DIM HX(1),HY(1)
100  DIM X(7),Y(7),B(7,7)
110  X(0)=-1:Y(0)=-2
120  X(1)=-2:Y(1)=-1
130  X(2)=-2:Y(2)=-1
140  X(3)=-1:Y(3)=2
150  X(4)=1:Y(4)=2
160  X(5)=2:Y(5)=1
170  X(6)=2:Y(6)=-1
180  X(7)=1:Y(7)=-2
190  C=0:GOSUB 250
200  C=1:GOSUB 250
210  C=1+(C=1)
220  GOSUB 390
230  IF F=-1 THEN 560
240  GOTO 210
250  X=0:Y=0:F=0
260  GOSUB 440
270  IF F=1 THEN PLAY"T1O3C2":F=0:GOTO 260
280  IF T=8 THEN RETURN
290  IF S=4 THEN Y=Y+1:IF Y>7 THEN Y=7
300  IF S=8 THEN Y=Y-1:IF Y<0 THEN Y=0
310  X=X+(S=1)-(S=2)
320  X=-X * (X>0)+(X>7)
330  GOTO 260
340  GOSUB 440:F=0
350  IF T=8 THEN RETURN
360  IF S=0 THEN S=4
370  IF S=8 THEN N=N-1:IF N<0 THEN N=7
380  IF S=4 THEN N=N+1:IF N>7 THEN N=0
390  X=HX(C)+X(N):Y=HY(C)+Y(N)
400  F=F+1:IF F>8 THEN F=-1:RETURN
410  IF X<0 OR X>7 OR Y<0 OR Y>7 THEN 360
420  IF B(X,Y)=1 THEN 360
430  GOTO 340
440  SPRITE C,136-16 * X,16 * Y+47
450  T=STRIG(C):S=STICK(C)
460  IF (S+T)=0 THEN 450
470  IF T<>8 THEN 540
480  IF B(X,Y)=1 THEN F=1:RETURN
```

```
490 B(X,Y)=1
500 HX(C)=X:HY(C)=Y
510 SPRITE C+2,136-HX(C)*16,16*HY(C)+47
520 LOCATE 15-2*HX(C),3+2*HY(C)
530 PRINT"*":PLAY"T1O3CDEG"
540 SPRITE C
550 RETURN
560 END ROUTINE
570 LOCATE 3,20:PLAY"T1O3CDET2 O4EGAC"
580 IF C=1 THEN PRINT"BLUE";
590 IF C=0 THEN PRINT"RED";
600 PRINT"WIN!!":END
```

背景

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0																												
1		8		7		6		5		4		3		2		1				160	J30	J30	J30	J30	J30	J30	I70	
2	K72	K52	K22	K52	K22	K52	K22	K52	K22	K52	K22	K52	K22	K52	K22	K52	L02			J20	K	N	I	G	H	T	J20	
3	K62		K62		K62		K62		K62		K62		K62		K62		K62	A		J00	J30	J30	J30	J30	J30	J30	J10	
4	K42	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K32											
5	K62		K62		K62		K62		K62		K62		K62		K62		K62	B										
6	K42	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K32											
7	K62		K62		K62		K62		K62		K62		K62		K62		K62	C										
8	K42	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K32											
9	K62		K62		K62		K62		K62		K62		K62		K62		K62	D										
10	K42	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K32											
11	K62		K62		K62		K62		K62		K62		K62		K62		K62	E										
12	K42	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K32											
13	K62		K62		K62		K62		K62		K62		K62		K62		K62	F										
14	K42	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K32											
15	K62		K62		K62		K62		K62		K62		K62		K62		K62	G										
16	K42	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K02	K52	K32											
17	K62		K62		K62		K62		K62		K62		K62		K62		K62	H										
18	L12	K52	K12	K52	K12	K52	K12	K52	K12	K52	K12	K52	K12	K52	K12	K52	L22											
19																												
20																												

〈跳马〉

将棋子放入棋盘，以跳棋方式移动，先不能移动者为失败。

〈游戏方法〉

2 人对战，用十字键改变移动场所，场所定了，用 A 钮来定位。

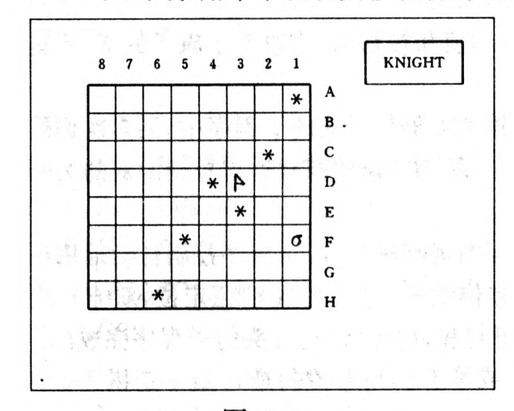


图 5.3

	S1		S8	
S2				S7
		□		
S3				S6
	S4		S5	

程序例1分析：

本程序是在棋盘中进行跳马。两人对玩，最终无路可走者为输。

根据跳马的规则，棋子的每一步最多有8个位置可走，如右图所示用S1—S8表示。这8个位置相对于棋子所在位置的座标分别为S1(－1，－2)，S2(－2，－1)，S3(－2，1)，S4(－1，2)，S5(1，2)，S6(2，1)，S7(2，－1)，S8(1，－2)。本程序第100语句至第180语句中定义的X数组和Y数组配对用来表示这8个相对位置座标。

另外，如果某一位置已有棋子走过，则在该位置上显示一个"＊"符号，以后程序可自动识别，此位置双方都不可再走。第100语句中定义的B数组用来记忆某一位置是否走过，如果B(n，n)＝1(n代表X，Y座标，其值为0—7)则表示该位置已走过；如果B(n，n)＝0，则表示该位置未走过，可以走。

本程序用小旗子作为标志，指示棋子可以走的位置，如果按操纵器＋字按钮的上方位时，则围绕棋子顺时针方向用小旗子指示棋子可以走的位置S1—S8，已走过棋了的地方，即显示有"＊"的位置会自动跳过，每按一次，移动一次。如果按＋字按钮的下方位时，则反过来逆时针方向用小旗子指示棋子可走的位置。A按钮是定位按钮，当用＋字按钮选定了要走的位置后，按一个A按钮，棋子就会移动到小旗子所指示的位置，原来棋子所在位置显示"＊"标记。然后小旗子会移动到对方棋子周围S1--S8位置中的某一个，由对手用另一个操纵器控制走棋。

下面进行程序分析：

程序的第20—70语句定义棋子、小旗子和它们的颜色。棋子有两种颜色，小旗小也有两种颜色，游戏的双方使 用不同的棋子和小旗子。

开棋时，小旗子出现在棋盘的右上角，游戏者通过按操纵器＋字按钮的上、下、左、右方位来分别控制小旗子上、下、左、右移动，每按一次，则在该方位上移动一格，以此来选择棋子的落位。此时，若按下A按钮，在所选定的位置上就会出现棋子，小棋了回到棋盘右上角的位置，由游戏的对方来选择第一步棋子的落位。这个功能是由程序的第190语句，第200语句、以第250语句开头的子程序和以第440语句开头的子程序共同完成的。

首先介绍一下以440语句开始的子程序，这个子程序主要完成定位功能。

在本程序中，调用这个子程序时C总是等于0或1，所以440语句的功能就是使小旗子根据X，Y的值出现在屏幕上，第450语句从操纵器上得到方向值和按钮值(参见STRIG指令和STICK指令)，如果按钮A按下，则令B(X，Y)＝1，记下此位置已有棋子走过，然后在第510语句让棋子移动到新的位置，第520—530语句在棋子的位置上显示"＊"标记。

第250语句开头的子程序是根据从操纵器上得到的方向值和按钮值，来改变小旗子的X，Y座标。

这样第190语句通过调用250语句开头的子程序(在第250语句开头的子程序中，又多次调用440语句开头的子程序)控制了游戏的一方第一步棋的定位。第200语句同样道理控制游戏的另一方第一步棋的定位。

在游戏双方第一步棋走过之后，游戏将按照前面所叙述的规则进行，即由程序控制自动地用小旗子提示可以走的位置，游戏者通过按＋字按钮的上、下方位键来选择，用A按钮定位，双方轮换走棋。这个功能是由程序的第210—240语句、340—430语句和以440语句开头的子程序完成的。

程序的第210语句—240语句轮换地使C变量等于0或等于1，用来控制是A选手走棋还是B选手走棋。在这段程序中调用390语句的子程序。

第390语句开始的子程序就是控制按照跳马的规则，自动选择S1—S8位置的。X(7)、Y(7)数

组配对(即 X(0),Y(0))、(X(1),Y(1))……)表示 S1—S8 位置相对于棋子位置的座标,第 390 语句即保证了小棋子的位置(X,Y)只能在棋子(HX(C),HY(C)周围的 SI—S8 的个位置内选择,第 360—380 语句根据从十字按钮输入的方向值,控制顺时针方向或逆时针方向来选择。第 400 语句用 F 变量作为计数器,如果 S1—S8 的 8 个位置都已有棋子走过,则令 F=-1,从子程序返回。在第 230 语句中判断若 F=-1 则游戏结束,跳转到第 560 语句,在屏幕上显示"BLUE WIN"(兰胜)或"RED WIN"(红胜)字样。

程序例 2:非凡的记忆力

程序清单:

```
10   WIEW:CGEN 2
20   MAX=5:I=Z:X=Y:C=0
30   PP$="CFGE"
40   Z$=CHR$(254):Z$=Z$+Z$+Z$+Z$+Z$
50   N$="          "
60   DIM PL(MAX),PX(3),PY(3),C(3)
70   PX(0)=16:PY(0)=8:C(0)=2
80   PX(1)=0:PY(1)=8:C(1)=4
90   PX(2)=8:PY(2)=0:C(2)=6
100  PX(3)=8:PY(3)=16:C(3)=8
110  '
120  PL(Z)=RND(4)
130  FOR I=0 TO Z
140  X=PX(PL(I)):Y=PY(PL(I))
150  C=C(PL(I))
160  PALETB 0,13,13,13,13
170  GOSUB 440
180  PLAY"T2O4"+MID$(PP$,PL(I)+1,1)+"3"
190  GOSUB 500
200  PALETB 0,13,&H16,&H27,2
210  PAUSE 10
220  NEXT
230  '
240  I=0
250  LOCATE 9,10:PRINT"YOU"
260  A$=INKEY$:IF A$="  " THEN 260
270  IF A$<CHR$(28) OR A$>CHR$(31) THEN PLAY"T1O1C1C1C1":GOTO260
280  IF(ASC(A$)-28)<>PL(I)THEN PLAY"T2O5C2R2F2R2E2":LOCATE 9,10:PRINT"
   ":GOTO 130
290  PALETB 0,13,13,13,13
300  X=PX(PL(I)):Y=PY(PL(I))
310  C=C(PL(I))
```

```
320 GOSUB 440
330 PLAY"T2O4"+MID$(PP$,PL(I)+1,1)+"3"
340 GOSUB 500
350 PALETB 0,13,&H16,&H27,2
360 I=I+1:IF I<=Z THEN 250
370 Z=Z+1:IF Z>MAX THEN410
380 LOCATE 9,10:PRINT"          "
390 PAUSE 50
400 GOTO 110
410 'END
420 CLS:CGSET 1,1:PRINT"GOOD!!"
430 END
440 FOR J=0 TO 5
450 LOCATE X,Y+J
460 PRINT Z$
470 NEXT
480 PALETB 0,13,C,C,C
490 RETURN
500 PALETB 0,13,13,13,13
510 FOR J=0 TO 5
520 LOCATE X,Y+J
530 PRINT N$
540 NEXT
```

背景

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0																												
1					K62	K62	K62								K62	K62	K62					I60	J30	J30	J30	J30	J30	I70
2					K62	K62	K62								K62	K62	K62					J20	S	U	P	E	R	J50
3					K62	K62	K62								K62	K62	K62					J20	M	E	M	O	R	Y
4		K52	K52	K52	L22	K62	K62								K62	K62	L12	K52	K52	K52		J00	J30	J30	J30	J30	J30	J70
5		K52	K52	K52	K52	L22	K62								K62	L12	K52	K52	K52	K52								
6		K52	K52	K52	K52	K52	L22								L12	K52	K52	K52	K52	K52								
7																												
8																												
9									K72	K52	K52	K52	L02															
10									K62				K62															
11									L12	K52	K52	K52	L22															
12																												
13																												
14																												
15		K52	K52	K52	K52	K52	L02								K72	K52	K52	K52	K52	K52								
16		K52	K52	K52	K52	L02	K62								K62	K72	K52	K52	K52	K52								
17		K52	K52	K52	L02	K62	K62								K62	K62	K72	K52	K52	K52								
18					K62	K62	K62								K62	K62	K62											
19					K62	K62	K62								K62	K62	K62											
20					K62	K62	K62								K62	K62	K62											

〈非凡的记忆力〉

计算机可以测验你的记忆力，看你是否记得住彩色绘图板上、下、左、右闪动的次序。

〈游戏方法〉

1人玩。首先，彩色画板会在上下左右4个方向任意闪烁，请记住顺序。画面中央若出现

“YOU”的话，则以相同的顺序按指示键。画面的上下左右各对应[←][→][↑][↓]。若中途发生错误，电脑将重来，再一次以相同的顺序闪烁。在程序的 20 行处决定闪烁次数的最大值，若值增大，闪烁次数会增加。

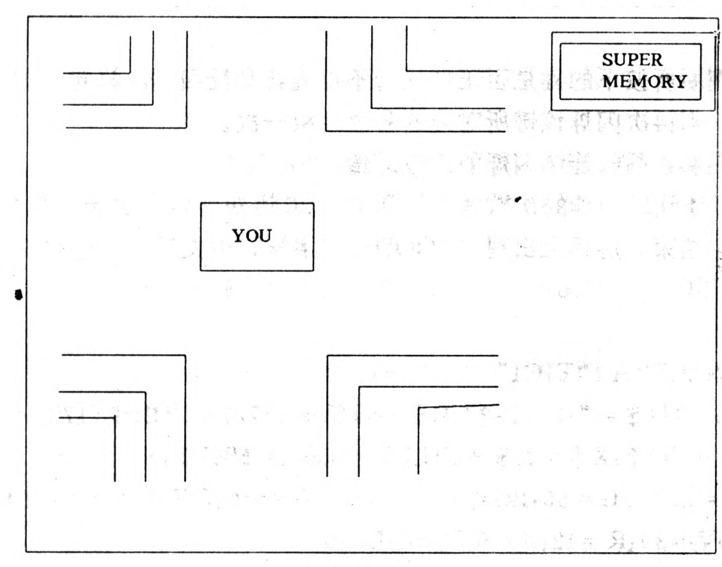


图 5.4

程序例 2 分析：

本程序可以测验游戏者的记忆力，彩色画板任意闪烁，游戏者按照彩色画板闪烁的顺序按方向键，画面的上、下、左、右分别对应着[←][→][↑][↓]四个按键。开始时，画面闪烁一次，然后屏幕中央出现“YOU”（中文意思“你”）字样，提示游戏者回答。若游戏者按键回答正确，则下次画面连续闪烁两次，然后再请读者回答。就这样连续闪烁的次数递增，最大值由程序第 20 语句中变量 MAX 决定。在按键回答的过程中，若每次按键回答正确，画板重复闪烁一次；若回答错误，画板将重新连续闪烁，然后再请游戏者回答。

程序的第 40 语句定义了 Z＄字符串，查看附录中的符号代码表 B 和封 3 符号图表 B 可知 Z＄字符串是由 5 个小方块顺序排列组成的长方条，这个长方条在后面第 440 语句开始的子程序中用来拼出闪烁用的彩色画板图案。

程序第 50 句定义 N＄字符中为 5 个空格，这个字符串在后面由第 500 句开始的子程序中，用来使彩色画板的图案消失，以达到闪烁的目的。

程序第 60—100 语句定义数组 PL、PX、PY、C。PX、PY 数组配对指定彩色画板的左上角在屏幕上的位置，从而指定整个彩色画板在屏幕上的位置。这四个位置的座标是（16，8)，(0，8)，(8，0）和（8，16)，分别对应着右、左、上、下四个位置。在这段程序中先用 PX、PY 数组定义好这四个位置座标，在后面控制画面闪烁时使用。C 数组用来定义彩色画板的颜色，在第 480 语句中使用。

程序第 120 语句—第 220 语句控制彩色画板闪烁，在这段程序中 Z 变量是控制连续闪烁次数的。第 120 语句 PL（Z）＝RND（4）是使 PL 数组为不可预测的数组。从 PL（0）到 PL（MAX）的值都是事先不可预知的，在第 140 语句中作为 PX、PY 数组的下标，即 PX（N）中的 N，指定彩色画板的定位位置（X，Y)，因此彩色画板闪烁的位置的顺序也就成为事先不可预测的了。在第 170 语句先调用 440 语句为首的子程序，然后再在第 190 语句调用 500 语句为首的子程序，前面子程序的作用是使彩色画板出现；后面子程序的作用是使彩色画板消失。这样使屏幕出

现彩色画板的闪烁。

当彩色画板闪烁结束后，第250语句就使屏幕上出现“YOU”字样提示游戏者按键回答。

第260语句从键盘取得所按下的键的键符，第270语句判断键符是不是[↑][↓][←][→]键符，若不是则重来。

第280语句判断所按下的键是否正确，若不正确，则反回130语句，重新连续地闪烁彩色画板一次。若正确，则再次闪烁该键所对应的彩色画板一次。

第360语句使彩色画板连续闪烁的次数递增。

第370语句判断连续闪烁的次数是否已到了第20语句所定义的最大值MAX，若已到最大值，则跳转到410语句结束，屏幕上出现“GOOD!!”字样，中文意思“好!!”

程序例3：飞碟UFO

程序清单：

```
10  VIEW:CGEN 2:PLAY"T1C1"
20  U0$="89:;":U1$="<=>?":B$=CHR$(172)+CHR$(173)+CHR$(174)+CHR
$(175):FOR I=0 TO 3:X$=X$+CHR$(180+I):NEXT:V=1
30  LG=16:RG=220:TG=50:BG=100:GX=0:GY=0:GV=16:BX=100:BY=220:BV=6:
DX=0:DY=0:DV=30:R=12:X=0:Y=0:D=0
40  G0$=CHR$(212):G1$=G1$(213):D0$=CHR$(209):D1$=G1$
50  PALETS 1,13,48,22,7:PALETS 2,13,48,22,1:PALETS 3,13,48,22,1:DEFSPRITE 1,
(2,1,0,0,0)=X$:DEF SPRITE 6,(3,1,0,0,0)=B$:GOSUB310:SPRITE ON
60  VX=V+RND(R):VY=V+RND(R)
70  C=RND(5):Y=Y+VY:X=X+VX*SGN(C):GOSUB260:T=STRIG(0):S=STICK(0)
80  SWAP G0$,G1$:SWAP D0$,D1$:SWAP U0$,U1$:GOSUB310
90  IF G=-1 THEN 120
100  IF T<8 THEN160
110  GX=BX+6:GY=BY:G=-1
120  GY=GY-GV
130  IF GY<TG THEN G=0:GX=0:GY=0:GOTO160
140  SPRITE 7,GX,GY
150  IF ABS(GX-X-5)<6 THEN IF ABS(GY-Y+8)<8 THEN330
160  IF S=1 THEN BX=BX+BV:IF BX>RG THEN BX=RG
170  IF S=2 THEN BX=BX-BV:IF BX<LG THEN BX=LG
180  IF D=-1 THEN210
190  IF ABS(BX-X)>5 THEN240
200  DX=X+6:DY=Y:D=-1
210  DY=DY+DV:IF DY>255 THEN SPRITE 5:D=0:DX=0:DY=0:GOTO240
220  SPRITE 5,DX,DY
230  IF (BY-DY)<6 THEN IF ABS(BX-DX+6)<8 THEN350
240  IF BG<210 THEN IF RND(4)=0 THEN TG=TG+V:BG=BG+V
250  GOTO70
260  IF X<LG THEN X=LG:VX=-VX
```

```
270  IF X>RG THEN X=RG:VX=-VX
280  IF Y<TG THEN Y=TG:VY=-VY
290  IF Y>BG THEN Y=BG:VY=-VY:IF BG>210 THEN360
300  RETURN
310  DEF SPRITE 0,(1,1,0,0,0)=U0$:DEF SPRITE 5,(0,0,0,0,0)=D0$:DEFSPRITE
7,(3,0,0,0,0)=G0$
320  SPRITE 0,X,Y:SPRITE 7,GX,GY:SPRITE 5,DX,DY:SPRITE 6,BX,BY:RETURN
330  FOR I=0 TO 9:SWAP X$,U0$:DEF SPRITE 0,(3,1,0,0,0)=U0$:SPRITE 0,X,Y:
PAUSE 10:PLAY"O5F1":NEXT
340  PR=PR+10:LOCATE 15,0:PRINT"SCORE:";PR:GX=0:GY=0:V=1+PR/20:GO-
TO30
350  FOR I=0 TO 5:SPRITE 6:PLAY"O1B1":PAUSE 8:SPRITE 6,BX,BY:PAUSE 10:
NEXT:SPRITE5:DX=0:DY=0:D=0:B=B+1:IF B<4 THEN30
360  PLAY"O1C4EC":INPUT"RETURN";I:RUN
```

背景

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0																												
1				M11	M31					G62			G52															
2		F41	D41	D41	D41	D41	F61									G62				G62					G52			
3			K12	K52	K52	K12							G62															
4		G52																			G62							G62
5																												
6				G62												G62												
7							G62				G62								G52					G62				
8																										G62		
9		G62																										
10																G52	G62						G62		G52			
11								G52																			G62	
12				G62																G62								
13													G62															
14																							G00	G10				
15									G00	G10										G00	G10	G00	G20	G30	G10			
16								G00	G20	G30	G10								G00	G20	G20	G20	G20	G30	G30	G10		
17	G40	M70	M70	G40	G40	M70	G20	G20	G20	G40	G30	G40	G40	M70	G40	M70	G40	G40	G40	G40	G20	G20	G40	G40	G30	G30	G40	G40
18	M70	G40	M70	M70	G40	G40	G40	G40	M70	M70	G40	G40	M70	G40	M70	M70	M70	G40	M70	M70	G40	G40	M70	G40	G40	G40	M70	M70
19	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32
20	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72

〈UFO〉

从谜一般的不明飞行物中出现了苍蝇，瞄准你的太空船，在苍蝇的攻击下，太空船将会躲藏起来，并开始还击。

〈游戏方法〉

1人玩。按1号控制器十字按钮的左、右方向移动太空船，按A按钮发射子弹攻击苍蝇。

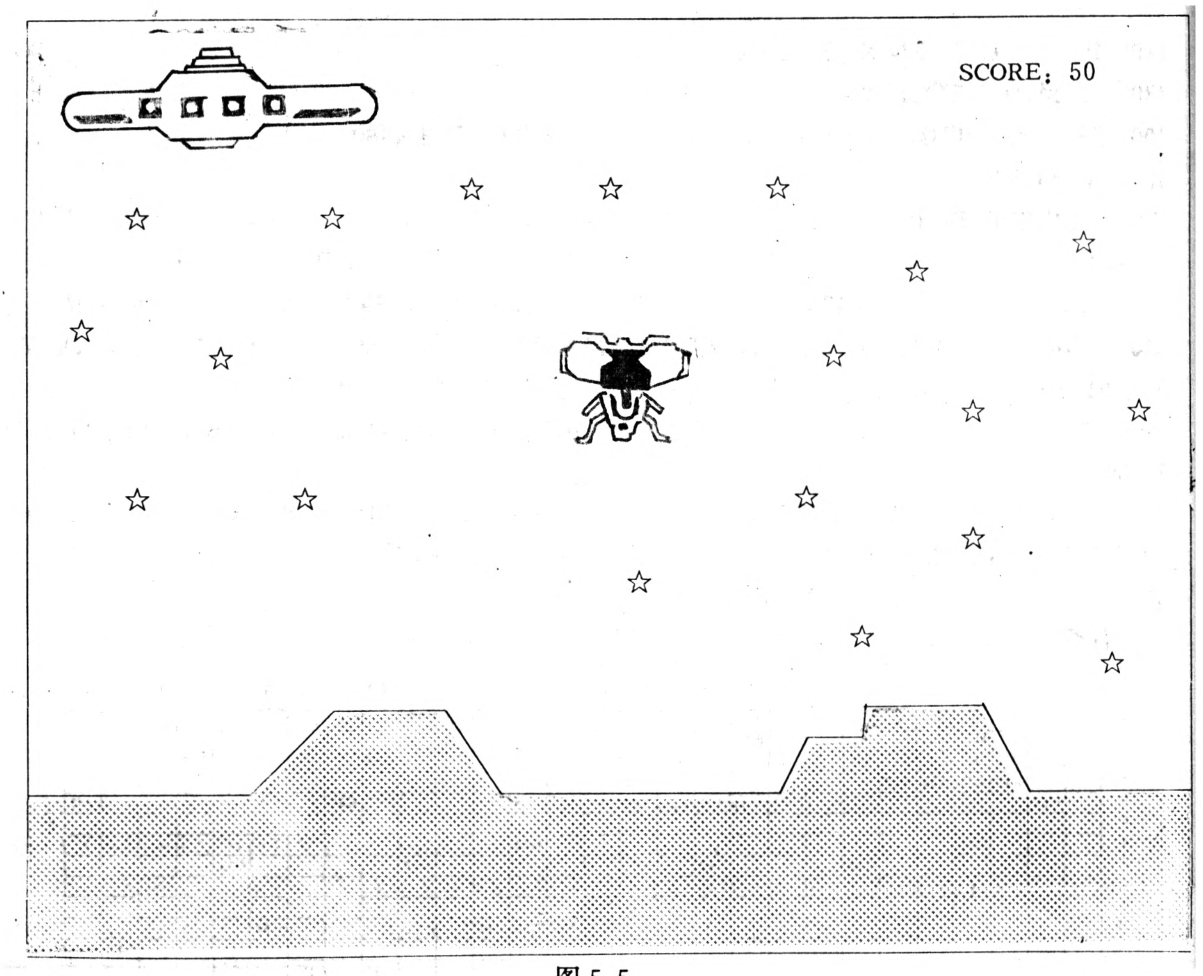


图 5.5

程序例 3 分析：

此程序是飞船打苍蝇游戏。游戏开始时，从屏幕左上角的不明飞行物 UFO 中飞出苍蝇，屏幕的下方出现太空船，苍蝇飞舞着翅膀上、下、左、右来回飞行，当苍蝇对准太空船时，就向太空船发射激光弹。游戏者用操纵器的＋字按键控制太空船左、右运动，用 A 按钮控制从太空船发射激光刀，向苍蝇进行还击。如果激光刀击中苍蝇，则苍蝇爆炸，在屏幕的右上角显示游戏者的累计得分；如果苍蝇发出的激光弹击中太空船，太空船闪烁 6 次，然后重新开始。如果四次命中太空船，则游戏结束，屏幕出现“RETURN”字样，中文意思是返回，等待游戏者按“RETVRN”键，游戏重新开始。

程序的第 20—40 语句是用来定义字符串和给一些变量赋值。

第 20 语句定义了字符串 U0＄，U1＄，B＄和 X＄。在后面的 DEF SPRITE 指令中分别用于定义卡通苍蝇 1，苍蝇 2，太空船和爆炸。

第 40 语句定义字符串 G0＄，G1＄，D0＄和 D1＄。在后面的 DEF SPRITE 指令中分别用于定义卡通激光刀 1，激光刀 2 和激光弹。

第 30 语句给一些变量赋值。LG 和 RG 分别定义苍蝇和太空船左、右运动的范围；TG 和 BG 定义苍蝇上、下运动的范围。GX 和 GY 是激光刀的座标；GV 是激光刀运动的速度。DX 和 DY 是激光弹的座标，DV 是激光弹的速度。BX 和 BY 是太空船的座标；BV 是太空船的速度。X 和 Y 地苍蝇的座标，它的运动速度是由第 60 语句的随机函数控制的。

第 50 语句，第 310 语句的 DEF SPRITE 指令定义卡通 0 为苍蝇，卡通 1 为爆炸，卡通 5 为激光弹，卡通 6 为太空船，卡通 7 为激光刀。

第 60 语句产生苍蝇 X 方向和 Y 方向的运动速度 VX 和 VY。

程序的主体是第 70 语句—第 250 语句的循环体。

第 70 语句首先调整苍蝇的位置座标 X、Y，并且调 260 语句的子程序，判断是否越出了上、下、左、右所限定的运动范围，如果超出了运动范围，则令其向相反方向运动。然后再从操纵器取得＋字键的方向代码 S 和按钮的控制代码 T。

第 80 语句交换字符串 G0＄和 G1＄、D0＄和 D1＄、U0＄和 U1＄，并且调用 310 语句的子程序，这个子程序的作用是定义卡通 0 苍蝇、卡通 5 激光弹、卡通 7 激光刀，并且使卡通 0 出现在（X，Y）位置、卡通 5 出现在（DX，DY）位置、卡通 6 出现在（BX，BY）位置，卡通 7 出现在（GX，GY）位置。交换字符串的原因是卡通 0 苍蝇、卡通 5 激光弹和卡通 7 激光刀都有两种姿势，两种姿势需要交替出现。

第 90 语句—140 语句控制激光刀的运动。当太空船发射出激光刀后，第 110 语句令变量 G＝－1，以后程序再次循环执行到第 90 语句时，则转到第 120 语句，使激光刀的 Y 方向座标递减，这样就使激光刀连续的向上运动。直到激光刀的 Y 座标等于 TY，第 130 语句令 G＝0，可以再次发射激光刀了。

第 150 语句判断激光刀是否击中苍蝇，若击中苍蝇，则跳到第 330 语句。

第 160—170 语句根据从操纵器＋字键输入的方向代码调整太空船的 X 方向座标 BX，控制太空船左右运动。

第 180—220 语句控制激光弹的运动，其原理同激光刀运动的原理。

第 230 语句判断激光弹是否命中了太空船，如果击中了太空船则跳转到第 350 语句，太空船发生闪烁，并且令记忆太空船被击中的次数的变量 B 加 1，如果 B＝4，即被击中 4 次，则游戏结束，屏幕出现“RETURN”字样，待游戏者按“ RETURN ”键，游戏重新开始。

第 250 语句控制程序跳转到 70 语句，使程序循环执行。

第 330 语句完成当太空船发射的激光刀命中苍蝇后令苍蝇发生爆炸，并且在屏幕右上角显示英文“SCORE”（中文意“分数”）安样和累计的分数，然后使苍蝇运动的速度增大。

程序例 4：66 号公路

程序清单：

```
10  VIEW：CGEN 2：CGSET 1，2：PALETS 0，13，22，39，2：SPRITE ON：PLAY"T1O3"
20  X1=150：X2=170：X3=190：X6=136：X7=208：C0＄=CHR＄（136）+CHR＄（137）
+CHR＄（138）+CHR＄（139）：C1＄=CHR＄（140）+CHR＄（141）+CHR＄（142）+CHR
＄（143）：B＄=CHR＄（214）
30  FOR I=4 TO 7：DEF SPRITE I，（0，0，0，0，0）=B＄：NEXT
40  F=2：SC=0：CA=3：L=0：X=208：Y0=150：V=0
50  L=L+1：M=L＊3+15：MV=M+33：D=（M+27）＊400
60  C=0
70  S=STICK（0）：T=STRIG（0）：V=V-1
80  IF S=1 THEN X=X+5
90  IF S=2 THEN X=X-5
100  IF T=8 THEN V=V+4：IF V>MV THEN V=MV
```

```
110  IF T=4 THEN V=V-4
120  IF V<1 THEN V=0
130  IF X<143 OR X>193 THEN 310
140  F=1
150  Y=Y0-V/2：DEF SPRITE 0，(0，1，0，0，0) =C0$：SPRITE 0，X，Y：SWAP C0
$，C1$
160  IF F1=-1 AND F2=-1 AND F3=-1 THEN SC=SC+1：F1=0：F2=0：F3=0
170  FOR I=1 TO3：DEF SPRITEI，(1，1，0，0，0) =C0$：NEXT
180  YY= (40-Y5) * (Y5>40)：SPRITE1，X1，Y1：SPRITE2，X2，Y2：SPRITE3，X3，
Y3：SPRITE4，X6，Y5：SPRITE5，X7，Y5：SPRITE6，X6，YY：SPRITE7，X7，YY
190  IF ABS (Y1-Y) <12 THEN F1=V>V1：IF ABS (X1-X) <10 THEN340
200  IF ABS (Y2-Y) <12 THEN F2=V>V2：IF ABS (X2-X) <10 THEN340
210  IF ABS (Y3-Y) <15 THEN F3=V>V3：IF ABS (X3-X) <10 THEN 340
220  Y1=Y1+V-V1：Y2=Y2+V-V2：Y3=Y3+V-V3：Y5=Y5+V：XX=XX+2：X1=
ABS (XX) +143：IF XX>48 THEN XX=-51
230  IF Y1>250 THEN V1=RND (M) +30：Y1=0
240  IF Y2>250 THEN Y2=0：X2=RND (58)：V2=RND (M) +30
250  IF Y3>250 THEN Y3=0：X3=RND (58) +138：V3=RND (M) +30
260  Y5=-Y5 * (Y5<250)：Y1=Y1- (Y1<0) * 255：Y2=Y2- (Y2<0) * 255：Y3=
Y3- (Y3<0) * 255
270  D=D-V：IF C-C/100 * 100=0 THEN GOSUB 380
280  IF D<0 THEN GOSUB380：PRINT"GOOD!"：PLAY"CGDAB"：SC=SC+ (500-C) /10：
GO TO 50
290  C=C+1：IF C<401 THEN70
300  PRINT"TIME UP"：PLAY"BAGFEDC"：F=3：L=L-1：GOTO 340
310  V=V-4：IF F=2 THEN V=V+4：IF V>MV-15 THEN V=MV-15
320  IF V<0 THEN V=0：F=2
330  GOTO 150
340  PLAY"BAEDC"：CA=CA-1：IF CA>-1 THEN X=208：V=0：GOSUB 380：ONFGO-
TO70，70，50
350  PRINT"GAME OVER!"：PRINT"TRY AGAIN"
360  IF STRIG (0) <>1 THEN 360
370  GOSUB 380：GOTO 10
380  IF SC>H1 THEN H1=SC
390  LOCATE 0，0：PRINT"HIGH SCORE"：H1：PRINT"LEVE"；L：PRINT"SCORE"；SC：
PRINT"CARS"；CA：PRINT：PRINT：PRINT"LEFT"；D/4；"M"：PRINT：PRINT"LEFT"；400-
C；"SEC"：PRINT：RETURN
```

背景

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
1															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
2															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
3															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
4															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
5															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
6															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
7															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
8															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
9															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
10															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
11															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
12															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
13															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
14															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
15															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
16															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
17															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
18															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
19															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		
20															M70	L40	M71	M71	M71	M71	M71	M71	M71	M71	L40	M70		

〈66 号公路〉

使比赛汽车飞驶在无尽 头的连续直线马路上。用加速器加速，快速地操纵把手，追赶挡住去路的干扰物。

〈游戏方法〉

1 人玩。在限定的时间内必须跑完所指定的距离，按操纵器＋字按钮的左右两边来操用把手，A 按钮是加速器，B 按钮是刹车，若与其它车子碰撞便会爆炸，在起动位置加速后，请进入跑道。

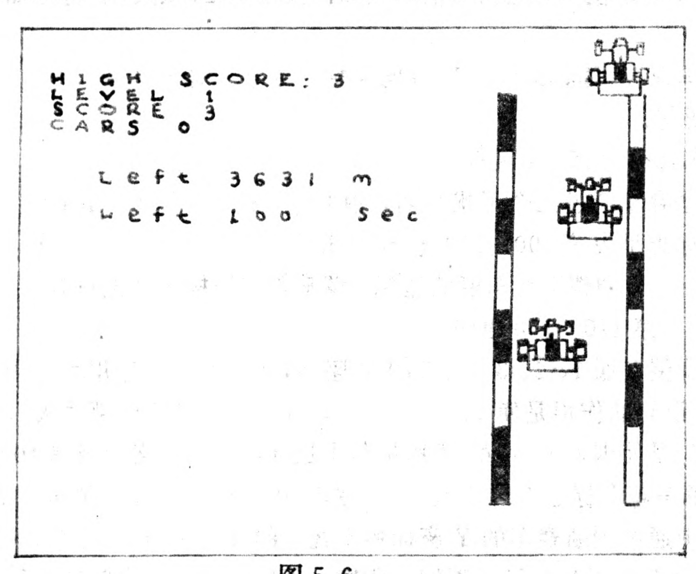


图 5.6

程序例 4 分析：

本程序是赛车游戏。游戏者必须在规定的时间内跑完全程。操纵器＋字按键的左、右方位可控制赛车左、右移动，躲避跑道上的其它干扰车辆，若碰上干扰车辆，赛车会发生爆炸。操纵器 A 按钮是加速，B 按钮是刹车。赛车共可与干扰车辆碰撞发生爆炸四次，超过四次，游戏结束。游戏的难度是分级的，难度级越高，赛车的最高速度越大，在规定的时间内所要跑完的路程越长。如果在规定的时间内，赛车跑完了全程，则难度级自动加 1，游戏重新进行；如果规定的时间用完，

赛车未跑完全程，则难度级不变，继续在这一难度级下重新开始。

本程序共定义了8个卡通，卡通0为赛车，卡通1、2、3为干扰车辆，卡通4、5、6、7为跑道边上的标志物。各卡通的位置座标分别用（X，Y），(X_1，Y_1）……（X_7，Y_7）表示。第30语句定义卡通4、5、6、7。第150语句定义赛车，其后面交换字符串变量C0＄和C1＄是为了使赛车的两个卡通姿 势轮流出现。第170语句定义卡通1、2、3。

程序第20语句中字符串变量C0＄，C1＄在第150语句和第170语句中是用来定义卡通赛车和干扰车辆的。字符串变量B＄在第30语句中用来定义跑道上的标志物卡通。

程序第40、50语句给一些变量赋值，其中L为难度级变量，MV为最高速度，D为在规定时间里所要跑完的距离。SC是用来累计得分的变量。CA是用来记录赛车碰撞爆炸的变量，初值为3，每碰撞一次CA变量减1，直到小于零，则游戏结束。X是赛车的X座标，Y0是赛 车的“基本座标”，在第150语句中令赛车的Y座标Y＝Y0－V/2，即赛车的Y座标为基本座标Y0减去V/2，V是速度变量。

第70—290语句是程序的主体，循环执行，完成以下几个任务

(1)．根据从操纵器取得的方向代码来调整赛车的X座标。第70—90语句。

(2)．根据按钮A、B是否按下来调整车速V。第100—120语句。

(3)．判断赛车是否越出跑道。第130语句。

如果寨车出了跑首，则从第130语句跳转到第310语句执行越出跑道的处理。第310语句令车速V减小4个单位，如果不按操纵器的十字按键使赛车返回跑道，则每次执行第310语句使车速V递减4个单位，直到车速减到零为止。此时第320语句令变量F＝2，当再次执行到第310语句，先令V减小4个单位（V已为负值），然后判断F＝2，则令V再增加4个单位，这样就保持了V＝0。

(4)．根据车速V调整赛车的Y方向座标，并使卡通0（赛车）出现在调整后的位置（X，Y）上。第150语句。

(5)．给赛车加分。第160语句。

加分的条件是赛车超过三个干扰车辆，用F1、F2、F3等于－1来作为超过干扰车辆1、2、3的标志，这三个标志是由第190—210语 句给出的。

(6)．根据车速V调整干扰车辆的座标，调整跑道边标志物的座标，并令它们出现在屏幕上。第220—260语句；第170—180语句。

第220语句调整卡通1、2、3、4、5的Y座标，XX变量的作用是使卡通1在跑道上左、右晃动。第230—260语句的作用是使1、2、3、4、5号卡通从屏幕底部消失时，再从屏幕上边跑道上出现。第180语句是使卡通6、7的Y座标与卡通4、5的Y座标相差40个单位。

(7)．给出赛车超车标志F1、F2、F3和判断赛车与干扰车辆碰撞。第190—210语句。

在这段程序中通过判断赛车的Y座标和干扰车的Y座标来判断赛车是否超过干扰车辆。同时比较X座标判断赛车与干扰车辆是否发生碰撞。若发生碰撞，则跳转到第340语句，令CA变量减1，若碰撞发生四次，则游戏结束，屏幕出现“GAME OVER”（游戏结束）“TRY AGAIN”（请再试）字样等待游戏者按START键重新开始游戏。

(8)．断是否跑完全程。第270—280语句。

第270语句使记录剩余路程的变量D递减，并且每隔100个时间单位，调用380语句的子程序，显示最高分、难度级、得分、赛车个数、剩余距离、剩余时间等参数。

第280语句，判断是否已跑完全程，若跑完全程，显示“GOOD!”（好!）字样，计算得分SC，

显示各个游戏参数。跳转到50语句，使难度级加1，继续重新开始游戏。

(9)．判断时间是否用完。第290—300语句。

在本句中，令时间变量C加1，如果C小于400，返回70语句继续。如果C大于400，显示"TIME UP"（时间到）字样，跳转到第340语句。判断4个赛车是否都用完了，如果未用完，赛车返回起跑位置，继续进行游戏。若用完了，游戏就结束了。

第380—390语句用于在屏幕上显示各个参数。

下面解释一下屏幕上出现的英文的意思：

HIGH SCORE	高分
LEVEL	难度级
SCORE	得分
CARS	车
LEFT	剩余
SEC	秒
M	米

程序例5：打字专家

程序清单：

```
10  CGEN 3：CGSET 1，1
20  VIEW：PLAY"T1O4C1"
30  PO=PP：I=J：K=L
40  PO$="O1O2O3O4O5"
50  PP$="CDEFGAB"
60  A$=B$
70  M3$="CANA!"
80  N$=CHR$（227）：NN$=N$+N$+N$+N$+N$+N$+N$
90  DIM DP（64），PP（64）
100  FOR I=0 TO 63：READ DP（I）：NEXT
110  DEF SPRITE 0，（0，0，0，0，0）=CHR$（199）
120  SPRITE ON
130  VIEW K=0：LOCATE 2，5：FOR I=9 TO 0 STEP -1：PRINT I;：NEXT
140  PAUSE 100：A$=CHR$（48+RND（43））：SPRITE 0
150  LOCATE 3，10：PRINT"THIS..."；
160  LOCATE 13，10：PRINT A$
170  GOSUB 460
180  FOR I=0 TO 75
190  FOR J=0 TO 5：B$=INKEY$
200  SPRITE 0，38+2*I，53
210  IF B$<>""THEN SWAP A$，B$：GOSUB 370：SWAP A$，B$：IF A$=B$ THEN
I=500：J=I
220  NEXT：NEXT：LOCATE 5，3：PRINT NN$
230  IF I>100 THEN PLAY"O4B1AG2FE3D4C"：K=K+1：GOTO 140
```

```
240  LOCATE 4, 15: PRINT K;"RIGHT ON."
250  LOCATE 3, 16: PRINT"TRY AGAIN?";: A$=INKEY$ (0): IF A$="N"THEN LOCATE 0, 18: END
260  GOTO 130
270  '
280  DATA 0, 21, 22, 23, 24, 24, 25, 25, 26, 27
290  DATA 14, 14, 5, 28, 6, 7
300  DATA 29, 21, 22, 23, 24, 24, 25, 25, 26, 27
310  DATA 14, 14, 5, 28, 6, 7, 21
320  DATA 7, 3, 2, 9, 16, 10, 10
330  DATA 11, 19, 11, 12, 13, 4, 4
340  DATA 20, 21, 14, 17, 8, 17, 18
350  DATA 3, 15, 1, 18, 0
360  DATA 0, 28, 0, 28, 7
370  IF A$<""THEN PLAY"O1C1EC": A$=" "
380  IF A$>"—" THEN PLAY"O5B1AB": LOCATE 5, 3: PRINT M3$: A$=" "390  IF A$=""THEN RETURN
400  IF A$="F"OR A$="J"THEN PLAY"O0A1: O0B1": GOTO 450
410  PO=DP (ASC (A$) -32) /7
420  PP=DP (ASC (A$) -32) -PO*7
430  PLAY MID$ (PO$, PO*2+1. 2) +MID$ (PP$, PP+1, 1)
440  PLAY MID$ (PP$, PP+1, 1)
450  RETURN
460  FOR J=0 TO 5: GOSUB 400: NEXT: RETURN
```

背景

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0		K72	K52	K52	K52	K52	L02																					
1		K62	T	Y	P	E	K62																					
2		L12	K52	K52	K52	K52	L22																					
3		160	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	I70				
4		J20																						J20				
5		J20																						J20				
6		J00	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J10				
7																												
8																												
9		I60	J30	J30	J30	J30	J30	J30	J30	J30	J30	I70	162	J32	I72													
10		J20										J20	J22		J22													
11		J00	J30	J30	J30	J30	J30	J30	J30	J30	J30	J10	J02	J32	J12													
12																												
13																												
14		I63	J33	J33	J33	J33	J33	J3	J33	J33	J33	J33	J33	J33	J33	J33	I73											
15		J23															J23											
16		J23															J23											
17		J03	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J13											
18																												
19																												
20																												

〈打字专家〉

记住字盘的排列是很困难的，如果 能幸运地记住 这些字母在键盘上的位置，就掌握了打字机，成为一各出色的打字员。

〈游戏方法〉

1人玩。在键盘字符区找出和屏幕中央部分右侧显示的字母、数字或符号相同的字符并输入之。当旗子右移消失后，时间也就结束了。

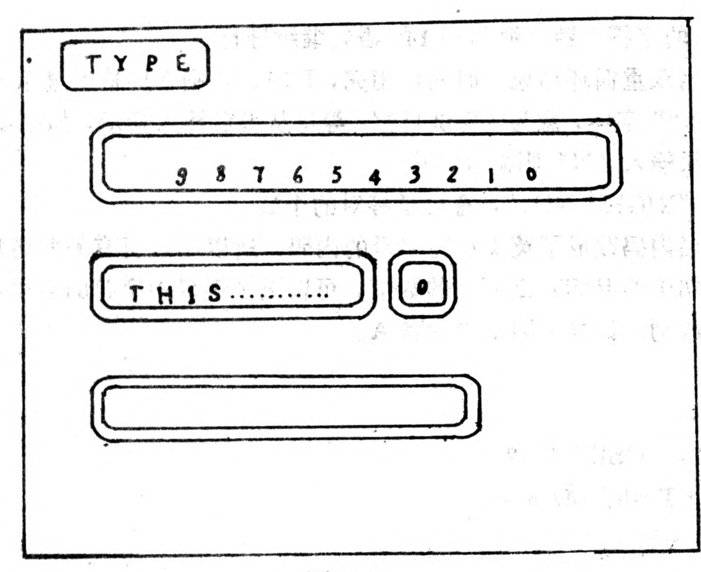


图 5.7

程序例 5 分析：

本程序可以帮助游戏者熟悉键盘。游戏开始，屏幕上方的方框中出现从 9—0 共 10 个数字，在数字 9 上方出现一个小旗子。在屏幕中央左边的方框中出现“THIS”（中文意思“这个”）提示符后，右边的方框随即出现一个随机的（任意的、且不可预测的）字符，然后小旗子即开始由左向右移动，指示时间。游戏者应在小旗子移到最右边的数字 0 的上方之前，在键盘上找到相同的字符输入。如果输入正确，则会出现新的字符，重复进行。如果时间用完了，也未能从键盘输入正确的字符，则显示前面正确输入字符的个数，游戏结束。

程序第 40、50、90 语 句定义的变量 PO＄、PP＄和数组 DP（64）、PP（64）与第 400—460 语句结合起来用于程序中演奏音乐。

程序第 80 语句定义了 NN＄变量，用于当时间用完后，在屏幕上显示出两条线。

第 110 语句定义卡通 0 为小旗子，用于指示时间。

第 130 语句在屏幕上方的方格中从左至右显示 9—0 10 个数字。

第 140—260 语句是程序的主体。

第 140 语句产生一个随机的字符 A＄，其范围是附录符号代码 B 中从 0—Z 共 43 个字符。

第 150 语句在屏幕中央左方的方框中显示提示符“THIS”。

第 160 语句在屏幕中央右方的方框中显示字符 A＄。

第 170 语句调用伴奏子程序。

第 180—220 语句是一个双重循环体。

在这个循环体中，第 200 语句用外重循环的控制变量 I 控制小旗子的 X 座标，使小旗子右移。

第 190 语句扫瞄键盘，并把从键盘输入的字符代入 B＄中。

第 210 语句判断如果有键按下，交换 A＄和 B＄，调用 370 语句开始的子程序。在这个子程序中再判断如果输入的键值小于空格的键值，那么让 A＄等于空格；如果输入的键值大于“—”的

键值，那么就是日文假名，在屏幕上显示“CANA”字样，中文意思是“假名”。并让A$等于空格。从370语句的子程序返回后，再次交换A$和B$，然后判断A$是否等于B$，如果相等，令I=J=500，跳出循环体。

第230语句判断循环控制变量I是否大于100，若I>100，则输入的字符是正确的，令K=K+1，记录正确输入的字符个数，返回到140语句继续进行。

第240语句是当双重循环结束，时间已用完，I<100，则在屏幕上显示K值。第250语句显示“TRY A-GAIN?”字样，意为“再试吗?”，等待从键盘输入指示，如果输入非“N”字符，则游戏重新开始；如果输入“N”则游戏结束。

第240句中的“RIGHT NO。”，意思是答对的个数。

由于把部分假名内码改成了英文小写字母的内码，所以不是所有的假名都能显示出来。

改动第140语句中的RND (X) 中的数值，可以限制键盘中字符的多少，也可使假名不显示出来。关于数值的改动，请参见附录中附表A。

程序例6：乌龟

程序清单：

```
10 CLS: CGEN 2: CGSET 1, 2
20 I=K: A=J: BT=5: MAX=3
30 PP=PR: W=-1
40 PLAY"O4T1"
50 P$="C2D2E2F2G2A2B2C2"
60 DIM AX (4), AY (4), Q (4)
70 DIM P (5, 1), C (5)
80 INPUT"PEOPLES"; PL
90 IF PL=0 OR PL>5 THEN PLAY"C1CC": CLS: GOTO 80
100 VIEW
110 LOCATE 13, 0: PRINT"TOR."
120 LOCATE 13, 2: PRINT"MUL."
130 FOR I=0 TO 4
140 Q (I) =RND (MAX) +BT
150 C (I) =MAX+BT-Q (I)
160 DEF SPRITE I, (0, 1, 0, 0, 0) =CHR$ (184) +CHR$ (185) +CHR$ (186) +
CHR$ (187)
170 LOCATE 17+2*I, 0: PRINT I+1
180 LOCATE 17+2*I, 2: PRINT C (I)
190 AX (I) =220
200 P (I, 0) =-1
210 NEXT
220 '
230 FOR I=0 TO PL-1
240 LOCATE 0, 21: PRINT I+1;"PLAYER NO。"
250 INPUT"TOR. NO. ?", M
```

```
260 IF M=0 OR M>5 THEN PLAY"C1CC": GOTO 240
270 P (I, 0) =M-1
280 NEXT
290 FOR M=4 TO 20: LOCATE 3, M: PRINT CHR$ (238): NEXT
300 FOR M=0 TO 4: LOCATE 25, 6+3*M: PRINT M+1: NEXT
310 '
320 SPRINT ON
330 FOR I=0 TO 4
340 A=RND (RND (Q (I)))
350 AY (I) =71+24*I
360 AX (I) =AX (I) -A*2
370 IF AX (I) <0 THEN AX (I) =0
380 PLAY MID$ (P$, A*2+1, 2)
390 SPRITE I, AX (I), AY (I)
400 IF AX (I) <50 THEN GOSUB 560
410 NEXT
420 K=K+1
430 IF K=20 OR K=50 THEN GOSUB 450
440 GOTO 310
450 '
460 FOR I=0 TO 4
470 LOCATE 10, 10: PRINT"CHANGE!"
480 A=RND (8)
490 IF A=7 THEN Q (1) =9: GOTO 530
500 IF A=6 THEN IF RND (5) =4 THEN Q (I) =0: GOTO 530
510 IF A<5 THEN 530
520 Q (I) =11-Q (I)
530 NEXT
540 LOCATE 10, 10: PRINT "          "
550 RETURN
560 '
570 IF I=W THEN RETURN
580 IF W < > -1 THEN 600
590 W=I: RETURN
600 '
610 PLAY"T6CEDFEGC2T1"
620 LOCATE 10, 6: PRINT"WIN"; W+1;"—"; I+1
630 SPRITE OFF
640 FOR J=0 TO PL-1
650 PR=0
```

```
660  IF P (J, 0) =W THEN PR=2
670  IF P (J, 0) =I THEN PR=1
680  PP=PR * C (P (J, 0))
690  LOCATE 9, 9+J * 2: PRINT J+1;"NO."; PP;"SCORE"
700  P (J, 1) =P (J, 1) +PP
710  NEXT
720  LOCATE 16, 22: PRINT "TRY AGAIN?";: A$ =INKEY$ (0)
730  IF A$ ="N"THEN 750
740  RUN
750  CLS: LOCATE 5, 10: PRINT"------END------": PLAY"CDGAC": END
```

背景

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50	D50
1	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52	D52
2	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42	G42
3	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52
4	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52	G52
5	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K52	K52	K52	K22
6				J20																								
7	1			J20																								
8	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K52	K52	K52	K02
9				J20																						J22		K62
10	2			J20																						J22		K62
11	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K52	K52	K52	K02
12				J20																						J22		K62
13	3			J20																						J22		K62
14	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K52	K52	K52	K02
15				J20																						J22		K62
16	4			J20																						J22		K62
17	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K52	K52	K52	K02
18				J20																						J22		K62
19	5			J20																						J22		K62
20	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K52	K52	K52	K12

〈乌龟〉

五只乌龟比赛，冲向终点目标，虽然倍率低的乌龟是优胜的候选者，但有时也会出现反败为胜的局面。紧张的情绪会一直持续到最后一刻。

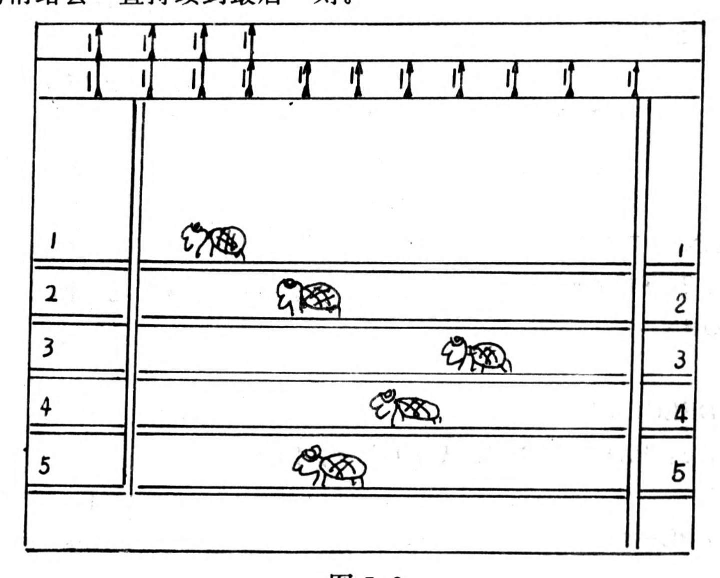


图 5.8

〈游戏方法〉

可能5人一起玩。首先要决定几个人玩，猜测哪个乌龟会最先到达终点，如果猜测正确，则根据其倍数率得分，执行中所显示的“CHANGE”是为使游戏更有趣和表示内部顺时针更动的信号。

程序例6分析：

这是一个乌龟赛跑的游戏。乌龟运动的速度是由第四章中介绍的随机函数产生的，因此是随机的，即不断变化的，且不可预先知道。程序开始，首先向游戏者询问有几个人参加（最多5个人），然后在屏幕右上角显示乌龟的号码和它们的倍率。在屏幕的左下角显示游戏者的号码，并等待游戏者从键盘输入所猜测的将取胜的乌龟的号码。待参赛的游戏者都输入完毕，游戏开始。倍率低的乌龟取胜的可能性最大，因为乌龟运动的速度是由随机函数产生的，因此也只能说可能 性最大，而不是倍率低的乌龟必然取胜。游戏过程中有时屏幕中央出现“CHANGE”字样，中文意思是“改变”，此时，控制乌龟运动速度的变量就改变了，这就使得结果变得更加捉摸不定，游戏更加有趣。

程序第80行中的“PEOPLES?”意为“几个人”，用于向游戏者询问有几个人参加游戏。

程序第110语句和第120语句中“TOR.”和“MUL.”。意为“乌龟”和“倍率”，这两个字出在屏幕右上角，用于表示乌龟的号码和它们的倍率，供游戏者猜测时参考。

程序第240语句和第250语句中的“PLAYER NO:”和：TOR. NO.?”。意为游戏者号码和乌龟号码。这两个字出现在屏幕左下角，游戏者号码由程序自动给出，乌龟号码由游戏者参考屏幕右上角乌龟的倍率输入。

程序第690语句中的“NO.”和“SCORE”，意为“第×只”和“得分”。供游戏结束时显示每只乌龟的得分。

整个程序分为五个部分：

第一部分从第80语句—210语句，这部分程序的作用是提示游戏者输入参加游戏的人数，产生每只乌龟的速度控制变量Q (I)和倍率C (I)，并在屏幕右上角显示乌龟号码和它们的倍率。第190语句令每只乌龟的X座标均为220，即程序开始时，使乌龟出现在起跑线上。

第二部分从第230语句一第300语句，这部分程序的作用是显示参赛的游戏选手号吗，等待游戏选手从键盘输入所猜测的乌龟的号码，并记忆于P数组中。第290和300语句分别用于显示终点线和起跑线处乌龟的号码。

第三部分从第320语句—第440语句，这部分程序的作用是控制乌龟的运动。这是一个循环程序，循环控制变量I从0—4，每次控制一只乌龟（第I只乌龟）的运动。乌龟运动的速度是由第340语句中所产生的A来控制的。A是由随机函数产生的，乌龟的速度控制变量Q (I) 越大，A的取值范围就越大，最终从总体效果来看，第I只乌龟运动速度快的可能性就大，也就是取胜的可能性应越大。第430语句使上面的循环每进行20次或50次，调用以450语句开头的子程序改变每只乌龟的速度控制变量Q (I)，增加游戏的趣味性。第400语句判断是否有乌龟到达终点，若有，则调用以560语句开始的子程序，判断游戏是否结束。

第四部分是从450语 句—550语句的子程序，它的作用是在游戏过程中在屏幕中央显示出“CHANGE”” 字样，然后改变每只乌龟的速度控制变量Q (I)。

第五部分是从560语句—750语句的子程序，它的作用是判断游戏结束，在屏幕上显示得胜的乌龟的号码和每只乌龟的得分。

第570语句—第590语句判断游戏是否结束，程序开始时第30语句给W赋值为−1，当第一

只乌龟到达终点时，在第590语句W=I，用W记忆第一个到达终点的乌龟的号码。当第2只乌龟到达终点时，在第570语句判断如果I≠W（I是第二个到达终点的乌龟的号码，W是第一个到达终点的乌龟的号码，所以I≠W），则在第580语句跳到第600语句进行结束处理。

第620语句使屏幕出现“WIN”字样（意“赢”）和第一个，第二个到达终点的乌龟的号码。第640语句—第710语句用于显示每位选手的得分。得分以倍率乘上一个系数而计算，第一名系数为2，第二名系数为1，其它名次系数为0。

第720—730语句向游戏者询问是否继续重新开始，屏幕上显示“TRY AGAN?”字样，意思是“重来吗?”如果键入“N”，则游戏结束。键入非“N”的任何键，则游戏重新开始。

程序例7：纸牌

程序清单：

```
10  VIEW：CGEN 3：CGSET 1，1：SPEITE ON
20  I=J：A=C：S=T：X=－1：Y=－1：ED=P：CX=XY：T0=T1：TT=0
30  N $=CHR $ (254)：N $=N $+N $+N $+N $：M0 $=CHR $ (243) +CHR $ (247)：M1 $=CHR $ (245) +CHR $ (248)
40  DIM D (5, 5)，PT (17)，PR (1)
50  FOR I=0 TO 5：FOR J=0 TO 5：LOCATE I*4+1，J*3+1：PRINT M0 $：LOCATE I*4+1，J*3+2：PRINT M1 $
60  A=RND (18)：IF PT (A) =2 THEN 60
70  D (I，J) =A：PT (A) =PT (A) +1：NEXT：NEXT
80  PALETS 0，13，&H12，&H22，2：PALETS 1，13，&H14，&H24，4：PALETS 2，13：&H16，&H26，6：PALETS 3，13，2，25，&H36
90  DEF SPRITE 0，(3，1，0，0，0) =N $：DEF SPRITE 5，(3，1，0，0，0) =N $
100  LOCATE 25，9：PRINT"L."：LOCATE 25，12：PRINT"R."
110  SPRITE 0，CX*32+24，CY*24+31：T0=－1：T1=－1：TT=7：GOSUB 170：SWAP TO，T1：X=CX：Y=CY：TT=6：GOSUB170
120  IF T0<>T1 THEN 150
130  PLAY"T204C1E1G1O5C6"：ED=ED+1：IF ED=18 THEN 300
140  PR (P) =PR (P) +10：LOCATE 23，10+P*3：PRINT PR (P)：SPRITE 6：SPRITE 7：GOTO 110
150  PLAY"O1T2D5E5C5"
160  D (X，Y) =T1：D (CX，CY) =T0：LOCATE CX*4+1，CY*3+1：PRINT M0 $：LOCATE CX*4+1，CY*3+2：PRINT M1 $：LOCATE X*4+1，Y*3+1：PRINTM0 $：LOCATE X*4+1，Y*3+2：PRINT M1 $：P=1－P：GOTO 110
170  SPRITE 5，216，95+24*P：PAUSE 10：SPRITE 5：S=STICK (P) T=STRIG (P)
180  IF S=1 THEN CX=CX+1：IF CX>5 THEN CX=0
190  IF S=2 THEN CX=CX－1：IF CX<0 THEN CX=5
200  IF S=4 THEN CY=CY+1：IF CY>5 THEN CY=0
210  IF S=8 THEN CY=CY－1：IF CY<0 THEN CY=5
220  IF S=0 THEN 240
230  SPRITE 0，CX*32+24，CY*24+31
240  IF T<8 THEN 170
250  SWAP D (CX，CY)，T0
260  IF T0=－1 THEN SWAP D (CX，CY)，T0：GOTO 170
270  SPRITE 0：PLAY"O5T1C1C1"：PAUSE 10：LOCATE CX*4+1，CY*3+1：PRINT"  "：LOCATE CX*4+1，CY*3+2：PRINT"  "
```

```
280  DEF SPEITE TT, (T0/6, 1, 1, 0, 0) =CHR $ (48+T0-T0/6 * 6): SPRITE TT, CX
* 32+24, CY * 24+31
290  RETURN
300  PRINT"TRY AGAIN !!"
310  T=STRIG (0): IF T=0 THEN 310
320  IF T=1 THEN RUN
330  IF T=2 THEN END
340  GOTO 300
350  END
```

背景

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0	K72	K52	K52	L02	K72	K52	K52	L02	K72	K52	K52	L02	K72	K52	K52	L02	K72	K52	K52	L02	K72	K52	K52	L02				
1	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62				
2	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62				
3	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32				
4	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62				
5	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62				
6	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32				
7	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62				
8	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62				
9	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	F72			
10	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62				
11	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62				
12	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	F72			
13	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62				
14	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62				
15	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32	K42	K52	K52	K32				
16	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62				
17	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62	K62			K62				
18	L12	K52	K52	L22	L12	K52	K52	L22	L12	K52	K52	L22	L12	K52	K52	L22	L12	K52	K52	L22	L12	K52	K52	L22				
19																												
20																												

〈纸牌〉

玩纸牌的关键要么完全记住翻过去的扑克的数码和花色，要么凭第六感观去取胜，你属于哪一类型？

〈游戏玩游〉

2 人玩。以操纵器＋字按钮来选择纸牌，以 A 按钮来翻牌，若翻开 2 张纸牌皆为同样花色、同样数字，则猜中了，可继续翻纸牌。翻错了的话则换对方玩

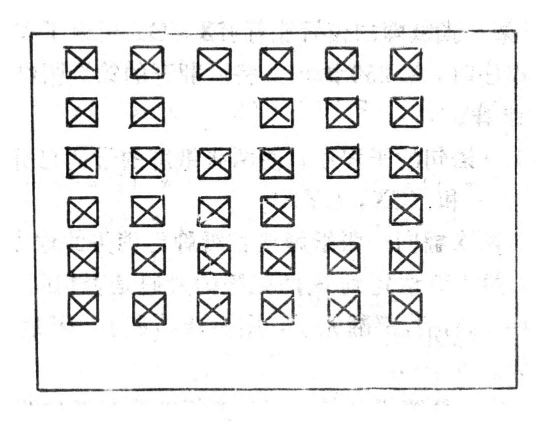


图 5.9

程序例 7 分析

此程序是翻纸牌游戏。两人玩，用操纵器十字按键移动光标来选择纸牌，用 A 按钮来翻牌。游戏开始，屏幕上出现 36 张纸牌的背面，游戏者翻 开两张后，若花色号码相同，计分加 10，可继续翻，且所翻的两张纸牌在屏幕上消失；若花色号码不相同，则恢复成纸牌的背面，此时，记忆力好的游戏者应记住此位置下纸牌的花色号码，以便下次轮到游戏时，可以准确地翻出两张相同的纸牌。纸牌翻完后，两人以屏幕右方显示的得分计胜负。

首先解释一下第 20—40 语句定义的变量和数组的程序中的用途。

第 20 语句定义的变量 X，Y 和 CX，CY 用于表示纸牌的座标。ED 用于记录已经翻对过的纸牌的对数，如果 ED=18，游戏则结束了。变量 P 等于 0 或 1 控制有效的操纵器是 1 号、还是 2 号。

第 30 语句定义的字符串变量 N$，用来定义卡通 0 和卡通 5，它们分别用于选择纸牌用的光标和在屏幕右方显示左或右操纵器有效的光标。M0$ 和 M1$ 用来拼出纸牌背面的图案。

第 40 语句定义和数组 D(5，5) 是一个 6×6 的数组，用来记忆 36 张纸牌的花色号码。第 50—70 语句程序自动给纸牌赋值时，PT (17) 数组用于控制相同纸牌只能有两张。PR (1) 数组有两个元素，PR (0) 用来记录操纵器 1 的得分，PR (1) 用来记录操纵器 2 的得分。

第 50—70 语句用于给纸牌赋值，使用 PT 数组控制相同号码的纸牌只能有两张。同时，本段程序使屏幕上出现 36 张纸牌的背面图案。

第 90 语句定义卡通 0 和卡通 5 两个光标。

第 100 语句在 36 张纸牌右方屏幕中央显示“L.”和“R.”字样（“左”、“右”），当光标在“L.”右边闪烁时，1 号操纵器有效；当光标“R.”右边闪烁时，2 号操纵器有效。

从第 110 语句开始到第 160 语句的循环体是程序的主体，控制着 1 号、2 号操纵器的选牌、翻牌、判断所翻的两张牌是否一样、计算得分和游戏是否结束。

第 110 语句的作用是两次翻纸牌。

开始时使光标出现在 CX，CY 所指示的纸牌的位置，令 T0=T1=−1，TT=7，调用 170 语句的子程序。

第 170 语句的子程序，根据十字按键输入的方位代码来调整光标的座标 CX，CY，移动光标选牌，如果 A 按钮按下则将该张纸牌翻过来。

第一次调用 170 语句的子程序时，将所选择的纸牌的号码在 250 语句交换给了 T0，并于第 270，280 语句在 CX，CY 所代表的位置将纸牌翻开。

从子程序返回后，交换 T0 和 T1，这样所翻开的第一张纸牌的号码就交换 给了 T1，而 T0=−1，然后再用 X，Y 记下第一张纸牌的位置座标 CX，CY 记录下来。

再调用 170 语句的子程序时，第 250 语句又将所翻开的第 2 张纸牌的号码交换给 T0，并于第 270，280 语句翻开第 2 张纸牌。

此时，经过两次调用 170 语句的子程序，使两张纸牌的号码已分别记于 T1 和 T0，两张纸牌的位置座标分别记于 (X，Y) 和 (CX，CY)。

另外，第 260 语句是对两次翻同一张纸牌或在纸牌已消失的地方翻牌的处理。

在两张纸牌都翻过后，第 120 语句判 断两张牌的号码是否相同。如果相同，首先再判断是不是所有纸牌都翻完了 (ED=18)，若都翻完了，则游戏结束了。然后给游戏者加分，并显示出来，最后使翻过的两张纸牌消失。

当两张纸牌不相同时，恢复两张纸牌的号码，恢复纸牌背面的图案，换另一个操纵器来选牌，这些处理是由第 160 语句完成的。

第 300 语句以后是结束处理。

第 300 语句在屏幕上显示“TRY AGAIN!!”（意“再试!”）字样。

然后等待从操纵器输入指令：按 START 则游戏重新开始；按 SELECT 键则游戏结束。

程序例 8：SCR $ 指令应用例

程序清单：

```
10  VIEW
20  PLAY"O4C1D1A1G1E1B"
30  SPRITE ON
40  CGSET 1, 0
50  PX=50: PY=56: MX=190: MY=150: DEF MOVE (0) =SPRITE (4, D, 1, 1, 0,
0): POSITION 0, PX, PY
60  DX=PX-MX: DY=PY-MY
70  IF ABS (DX) <8 AND ABS (DY) <8THEN 360
80  S1=-1* (DX>0) -2* (DX<0): S2=-4* (DY>0) -8* (DY<0)
90  IF ABS (DX) <ABS (DY) THEN SWAP S1, S2
100  S=S1: GOSUB 270: GOSUB 290
110  IF D<>0 THEN 140
120  SWAP S1, S2: S=S1: GOSUB 270: GOSUB 290
130  IF D=0 TNEN 160
140  DEF MOVE (1) =SPRITE (11, D, 1, 3, 0, 0): POSITION 1, MX, MY
150  MOVE 1: PIAY"O1 C1 C1 C1"
160  S0=STICK (0)
170  S=S0: GOSUB 280: GOSUB 290
180  IF D=0 THEN 250
190  DEF MOVE (0) =SPRITE (4, D, 1, 3, 0, 0)
200  POSITION 0, PX, PY
210  MOVE0: PIAY"O3B1D1"
220  XX= (PX+7) /8-2: YY= (PY+7) /8-3
230  IF SCR$ (XX, YY) =CHR$ (199) THEN LOCATE XX, YY: PRINT" ";: CN=CN
+1: LOCATE 10, 23: PRINT"SCORE"; CN;: PLAY"O4 C1 A1 G1"
240  IF MOVE (0) =-1 OR MOVE (1) =-1 THEN 240
250  PX=XPOS (0) =: PY=YPOS (0): MX=XPOS (1): MY=YPOS (1)
260  GOTO 60
270  X=MX- (S=1) *4+ (S=2) *4: Y=MY- (S=4) *4+ (S=8) *4: RETUN
280  X=PX- (S=1) *4+ (S=2) *4: Y=PY- (S=4) *4+ (S=8) *4: RETURN
290  C1= (X-1) /8-2: L1= (Y-1) /8-3
300  C2=X+16: C2= (C2-1) /8-2: L2=Y+16: L2= (L2-1) /8-3
310  D=-3* (S=1) 3* (SCR$C2, L1) =" ") * (SCR$ (C2, L2) =" ")
320  D=D-7* (S=2) *SCR$ (C1, L1) =" ") * (SCR$ (C1, L2) =" ")
330  D=D-1* (S=8) *SCR$ (C1, L1) =" ") * (SCR$ (C2, L1) =" ")
340  D=D-5* (S=4) *SCR$ (C1, L2) =" ") * (SCR$ (C2, L2) =" ")
```

```
350  RETURN
360  PLAY"O4 G1 C1 G1": FOR Q=0 TO 3: CGSETO, 0: CGSET1, 1: CGSET0, 0: NEXT
370  PLAY"O1 C1 G1 A1 C1 D1": CLS: SPRITE OFF
380  LOCATE 5, 10: PRINT"-------"END------": END
```

背景

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32
1	F32									F32	F32																	F32
2	F32			F72						F32	F32				F72				F72				F72					F32
3	F32																											F32
4	F32														F32	F32	F32	F32	F32	F32	F32	F32	F32					F32
5	F32								F72						F32								F32	F32				F32
6	F32				F32	F32							F32	F32					F72				F32	F32		F72		F32
7	F32				F32	F32							F32	F32														F32
8	F32																											F32
9	F32				F72									F72				F32	F32									F32
10	F32							F32	F32									F32	F32				F72					F32
11	F32	F32	F32	F32	F32	F32	F32	F32	F32																			F32
12	F32												F32	F32														F32
13	F32												F32	F32					F72							F72		F32
14	F32		F72								F72		F32	F32														F32
15	F32					F32	F32	F32					F32	F32									F32	F32				F32
16	F32					F32	F32	F32					F32	F32									F32	F32				F32
17	F32		F72																									F32
18	F32							F72			F72			F72				F32	F32					F72				F32
19	F32																	F32	F32									F32
20	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32

〈本例是使用 SCR $ 的例题〉(并非是游戏)

- 请先描绘背景再执行
- 以 ← → ↑ ↓ 键操纵企鹅
- 当小妖怪近企鹅时，企鹅想顺利地逃脱并通过有旗子的地方。
- 经过旗子时，操纵企鹅的中央部位，使其和旗子的上部重合。
- 乌龟或妖怪一旦碰到砖头，就不会向前走了。

在 BG 绘图程序的情况下，可以改变砖头或旗子的位置，或者变更符号，变更各种样式。

程序例 8 分析：

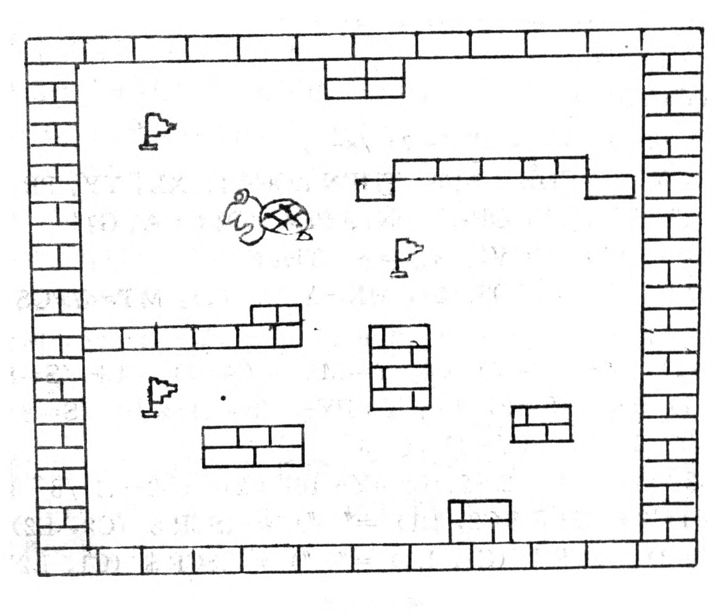


图 5.10

这是一个练习 SCR $ 指令的例子。屏幕的背景上有砖墙和小旗了，在程序的控制下小妖怪总是企图靠近企鹅，企鹅在操纵器的控制下躲避小妖怪，并去拔小旗。无论是企鹅还是小妖怪，前进时碰到砖墙就必须改变前进方向，才能继续运动。小妖怪若撞上企鹅，游戏就结束了。

第 50 语句定义变量。(PX，PY）是企鹅的座标，(MX，MY）是小妖怪的座标。

程序从第 60—150 语句，可分为两部分。

第一部分为 60—150 语句，控制小妖怪的运动。

在第 60 语句中令 DX＝PX－MX，即企鹅的 X 座标减去小妖怪的 X 座标；DY＝PY－MY，即企鹅的 Y 座标减去小妖怪的 Y 座标。根据 DX、DY 在第 70 语句判断企鹅和小妖怪是否撞上，若撞上，则跳转到到第 360 语句，游戏结束。

第 80—100 语句根据企鹅与小妖怪的相对位置，控制小妖怪的运动方向，令小妖怪追赶企鹅。

第 80 语句根据 DX 大或小于零，得到控制小妖怪向左或向右运动的控制变量 S1，同样道理根据 DY 得到控制小妖怪向上或向下运动的控制变量 S2。

第 90 语句决定如果企鹅与小妖怪在 X 方向上的距离差大于在 Y 方向上的距离差，则令小妖怪沿 X 方向前进，反之，则沿 Y 方向前进。

第 100 语句调用 2 个子程序来判断在小妖怪运动的前方，是否遇到了砖墙，若未遇到砖墙则返回时得到一个方向控制变量 D，这个方向控制变量在第 140 语句的 DEF MOVE 指令中控制小妖怪运动的方向。若遇到砖墙，则返回时得到的方向控制变量 D 等于零。

第 110 语句判断如果 D 不等于 0，即小妖怪运动的前方没有砖墙，则跳转到第 140 语句，令小妖怪沿 D 所规定的方向前进。如果 D＝0，即小妖怪运动方向上遇到了砖墙，则在第 120 语句中换一个方向再试试，如果返回的方向控制变量还是 0，那么小妖怪就暂时不动了。

第二部分是 160—250 语句，控制企鹅的运动。

在第 160 语句首先从操纵器＋字按键取得方向代码 S0，然后 调用 280、290 语句的子程序，判断企鹅运动的方向上是否遇到了砖墙，并且返回一个方向控制变量 D，同样，如果 D 等于 0，企鹅将不能按照预想的方向前进，如果 D 不等于 0，第 190 到 210 语句交控制企鹅沿 D 所代表的方向前进。

第 220—230 语句判断企鹅是否与小旗子重合，若重合则小旗子消失，屏幕上显示企鹅所拔小旗子的数量。

第 260 语句使程序循环进行。

第 290—350 语句的子程序的作用是将方向代码 S 转换为 DEF MOVE 指令中的方向控制变量 D，并且判断在卡通的运动方向上是否碰到了砖墙。

索引一　操作代码

FBASIC 除了通常的操作键以外，更具有许多容易操作的字键功能。

CTRL＋	处　理　内　容	参　　考
A	INS 状态的开关	
C	BREAK　但在程序执行中无效	
D	初始化　CGEN2、SPRINT OFF，CTR＋A 的解除。	
E	消去一行	
V	改变光标形状	
K	使光标回到左上角	HOME
L	清除画面	
R	与 INS 功能相同	
W	成为英文、数字状态	
Z	光标以后画面被清除	

索引二　存储区分配表

（16 进制）

地址	用途
&H0000 – &H07FF	RAM 工作区（在主机中）
&H0800 – &H1FFF	未　用
&H2000 – &H5FFF	系统使用
&H6000 – &H6FFF	未使用
&H7000 – &H77FF（含 &H703F）	RAM 工作区用，在 BS 卡中
&H7800 – &H7FFF	未使用
&H8000 – &HFFFF	程序 ROM 区

&H7000～&H703F 范围内请勿使用 POKE 指令，它被使用于系统中。

索引三　出错信息一览表

FBASIC 若在程序执行中发现错误，出错信息便会显示于画面上，停止程序的执行变成等待指令输入状态。

直接状态执行中，发生错误会显示出：

如　　? SN ERROR　　　　　　　　　；语法错误

程序状态执行中，发生错误会显示出：

如　　? SN ERROR IN 100　　　　　　；语法错误在 100 行

如果在程序执行中出现错误信息而不了解其原因时，请参阅下表。

错误表示	出　错　信　息	
NF	NEXT without FOR	有 NEXT 指令，无 FOR 指令
SN	Syntax error	语法错误
RG	RETURN without GOSUB	无 GOSUB 指令，有 RETURN 指令
OD	Out of DATA	READ 读的数据未备于 DATA 指令中。
IL	Illegal function call	非法功能调用
OV	Overflow	运算结果超过许可的范围。
OM	Out of memory	内存不够
UL	Undefined line Number	无 GOTO、GOSUB、IF 等 指令所指定的行号。
SO	Subscript out of range	变量下标超出范围。
DD	Duplicate Definition	定义重复。
DZ	Division by zero	用 0 作除数。
TM	Type mismatch	变量型式不一致。
ST	String too long	文字超过 31 个。
FT	Formula too complex	式子过于复杂，例如（　）太多。
CC	Can ’ t continue	由 CONT 无法再开始执行程序。
MO	Missing operand	未指定所需参数。
TP	Tape read ERROR	无法从磁带机正确读写文件。

索引四　符号代码表

下面的表 A 和表 B 给出了 FBASIC 的符号代码，FBASIC 中的指令 CHR＄（n）及 ASC（“A＄”）的字符代码转换全部对应此表。符号代码表 A 和 B 的符号、文字、记号在背景面或卡通面皆可使用，哪种画面使用哪种图表由 CGEN 指令指定。

▲符号代码表 A（主要用于卡通面）

请参阅本书彩色封底卡通图案 4 角的 10 进制数字。

代码十进制	代码十六进制	说明	代码十进制	代码十六进制	说明	代码十进制	代码十六进制	说明	代码十进制	代码十六进制	说明
0	00		32	20		64	40		96	60	
1	01	玛丽沃	33	21	丽 莎	65	41	飞 鱼	97	61	企 鹅
2	02	(走 1)	34	22	(走 2)	66	42	(1)	98	62	(走 1)
3	02		35	23		67	43		99	63	
4	04		36	24		68	44		100	64	
5	05	玛丽沃	37	25	丽 莎	69	45	飞 鱼	101	65	企 鹅
6	06	(走 2)	38	26	(走 3)	70	46	(左 2)	102	66	(走 2)
7	07		39	27		71	47		103	67	
8	08		40	28		72	48		104	68	
9	09	玛丽沃	41	29	丽 莎	73	49	飞 鱼	105	69	企 鹅
10	0A	(走 3)	42	2A	(跳)	74	4A	(左上 1)	106	6A	(正面)
11	0B		43	2B		75	4B		107	6B	
12	0C		44	2C		76	4C		108	6C	
13	0D	玛丽沃	45	2D	丽 莎	77	4D	飞 鱼	109	6D	企 鹅
14	0E	(跳)	46	2E	(滑)	78	4E	(左上 2)	110	6E	(背面)
15	0F		47	2F		79	4F		111	6F	
16	10		48	30		80	50		112	70	
17	11	玛丽沃	49	31	丽 莎	81	51	飞 鱼	113	71	火 球
18	12	(滑)	50	32	(背面)	82	52	(前 1)	114	72	(1)
19	13		51	33		83	53		115	73	
20	14		52	34		84	54		116	74	
21	15	玛丽沃	53	35	丽 莎	85	55	飞 鱼	117	75	火 球
22	16	(背面)	54	36	(跌下)	86	56	(前 2)	118	76	(2)
23	17		55	37		87	57		119	77	
24	18		56	38		88	58		120	78	
25	19	玛丽沃	57	39	苍 蝇	89	59	妖 怪	121	79	车
26	1A	(摔倒)	58	3A	(1)	90	5A	(1)	122	7A	(左 1)
27	1B		59	3B		91	5B		123	7B	
28	1C		60	3C		92	5C		124	7C	
29	1D	丽 莎	61	3D	苍 蝇	93	5D	妖 怪	125	7D	车
30	1E	(走 1)	62	3E	(2)	94	5E	(2)	126	7E	(左 2)
31	1F		63	3F		95	5F		127	7F	

代码十进制	代码十六进制	说明	代码十进制	代码十六进制	说明	代码十进制	代码十六进制	说明	代码十进制	代码十六进制	说明
128	80	车（左上1）	160	A0	杀手卫星（上）	192	C0	螃蟹（1）	224	E0	向上羽毛笔
129	81		161	A1		193	C1		225	E1	
130	82		162	A2		194	C2		226	E2	
131	83		163	A3		195	C3		227	E3	
132	84	车（左上2）	164	A4	太空船（左）	196	C4	螃蟹（2）	228	E4	
133	85		165	A5		197	C5		229	E5	
134	86		166	A6		198	C6		230	E6	
135	87		167	A7		199	C7		231	E7	
136	88	车（上1）	168	A8	太空船（左上）	200	C8	飞鸟（1）	232	E8	向下羽毛笔
137	89		169	A9		201	C9		233	E9	
138	8A		170	AA		202	CA		234	EA	
139	8B		171	AB		203	CB		235	EB	
140	8C	车（上2）	172	AC	太空船（上）	204	CC	飞鸟（2）	236	EC	
141	8D		173	AD		205	CD		237	ED	
142	8E		174	AE		206	CE		238	EE	
143	8F		175	AF		207	CF		239	EF	
144	90	太空船（1）	176	B0	爆炸（开始）	208	D0	激光弹（1）（2）	240	F0	1
145	91		177	B1		209	D1		241	F1	2
146	92		178	B2		210	D2	激光束（1）（2）	242	F2	3
147	93		179	B3		211	D3		243	F3	4
148	94	太空站（2）	180	B4	爆炸（散发）	212	D4	激光刀（1）（2）	244	F4	COMP-UTER
149	95		181	B5		213	D5		245	F5	
150	96		182	B6		214	D6	卡通面用代码	246	F6	
151	97		183	B7		215	D7		247	F7	
152	98	杀手卫星（左）	184	B8	乌龟走（1）	216	D8		248	F8	
153	99		185	B9		217	D9	音乐板用三音域（1）（2）（3	249	F9	
154	9A		186	BA		218	DA		250	FA	OPER-ATOR
155	9B		187	BB		219	DB		251	FB	
156	9C	杀手卫星（左上）	188	BC	乌龟立正（2）	220	DC	光标（1）	252	FC	
157	9D		189	BD		221	DD	光标（2）	253	FD	
158	9E		190	BE		222	DE	指示器（1）	254	FE	
159	9F		191	BF		223	DF	指示器（2）	255	FF	

▲符号图表 B（主要用于背景面）

键盘的文字、记号、BG 绘图程序所使用的符号。0—31 及 184—255 是背景面所用的各种图形，其余代表数字和字母符号。

代码 十进制	代码 十六进制	对应的 符号	代码 十进制	代码 十六进制	对应的 符号	代码 十进制	代码 十六进制	对应的 符号	代码 十进制	代码 十六进制	对应的 符号
0	00	(A0)	32	20		64	40	@	96	60	
1	01	(A1)	33	21	!	65	41	A	97	61	a
2	02	(A2)	34	22	•	66	42	B	98	62	b
3	03	(A3)	35	23	#	67	43	C	99	63	c
4	04	(A4)	36	24	$	68	44	D	100	64	d
5	05	(A5)	37	25	%	69	45	E	101	65	e
6	06	(A6)	38	26	&	70	46	F	102	66	f
7	07	(A7)	39	27	,	71	47	G	103	67	g
8	08	(B0)	40	28	(	72	48	H	104	68	h
9	09	(B1)	41	29	)	73	49	I	105	69	i
10	0A	(B2)	42	2A	*	74	4A	J	106	6A	j
11	0B	(B3)	43	2B	+	75	4B	K	107	6B	k
12	0C	(B4)	44	2C	,	76	4C	L	108	6C	l
13	0D	(B5)	45	2D	—	77	4D	M	109	6D	m
14	0E	(B6)	46	2E	.	78	4E	N	110	6E	n
15	0F	(B7)	47	2F	/	79	4F	O	111	6F	o
16	10	(C0)	48	30	0	80	50	P	112	70	p
17	11	(C1)	49	31	1	81	51	Q	113	71	q
18	12	(C2)	50	32	2	82	52	R	114	72	r
19	13	(C3)	51	33	3	83	53	S	115	73	s
20	14	(C4)	52	34	4	84	54	T	116	74	t
21	15	(C5)	53	35	5	85	55	U	117	75	u
22	16	(C6)	54	36	6	86	56	V	118	76	v
23	17	(C7)	55	37	7	87	57	W	119	77	w
24	18	(D0)	56	38	8	88	58	X	120	78	x
25	19	(D1)	57	39	9	89	59	Y	121	79	y
26	1A	(D2)	58	3A	：	90	5A	Z	122	7A	z
27	1B	(D3)	59	3B	；	91	5B		123	7B	
28	1C	(D4)	60	3C	<	92	5C	¥	124	7C	
29	1D	(D5)	61	3D	=	93	5D		125	7D	
30	1E	(D6)	62	3E	>	94	5E	ˆ	126	7E	
31	1F	(D7)	63	3F	?	95	5F	—	127	7F	

代　码 十进制	代　码 十六进制	对应的 符　号	代　码 十进制	代　码 十六进制	对应的 符　号	代　码 十进制	代　码 十六进制	对应的 符　号	代　码 十进制	代　码 十六进制	对应的 符　号
128	80		160	A0		192	C0	(F0)	224	E0	(J0)
129	81		161	A1		193	C1	(F1) 玛	225	E1	(J1)
130	82		162	A2		194	C2	(F2) 丽	226	E2	(J2) 迷
131	83		163	A3		195	C3	(F3) 兄	227	E3	(J3)
132	84		164	A4		196	C4	(F4) 弟	228	E4	(J4) 路
133	85		165	A5		197	C5	(F5) 用	229	E5	(J5)
134	86		166	A6		198	C6	(F6)	230	E6	(J6)
135	87		167	A7		199	C7	(F7)	231	E7	(J7)
136	88		168	A8		200	C8	(G0)	232	E8	(K0)
137	89		169	A9		201	C9	(G1) 风	233	E9	(K1)
138	8A		170	AA		202	CA	(G2)	234	EA	(K2)
139	8B		171	AB		203	CB	(G3) 景	235	EB	(K3) 图
140	8C		172	AC		204	CC	(G4)	236	EC	(K4) 表
141	8D		173	AD		205	CD	(G5) 星 (1)	237	ED	(K5) 边
142	8E		174	AE		206	CE	(G6) 星 (2)	238	EE	(K6) 线
143	8F		175	AF		207	CF	(G7) 球	239	EF	(K7)
144	90		176	B0		208	D0	(H0)	240	F0	(L0)
145	91		177	B1	。	209	D1	(H1) 金	241	F1	(L1)
146	92		178	B2	〔	210	D2	(H2) 刚	242	F2	(L2)
147	93		179	B3	〕	211	D3	(H3) 二	243	F3	(L3) 各
148	94		180	B4	c	212	D4	(H4) 代	244	F4	(L4) 板
149	95		181	B5	×	213	D5	(H5) 用	245	F5	(L5) 用
150	96		182	B6	÷	214	D6	(H6)	246	F6	(L6) 框
151	97		183	B7		215	D7	(H7)	247	F7	(L7)
152	98		184	B8	(E0)	216	D8	(10)	248	F8	(M0)
153	99		185	B9	(E1)	217	D9	(11)	249	F9	(M1) 图
154	9A		186	BA	(E2)	218	DA	(12) 岛	250	FA	(M2)
155	9B		187	BB	(E3)	219	DB	(13)	251	FB	(M3) 形
156	9C		188	BC	(E4)	220	DC	(14)	252	FC	(M4)
157	9D		189	BD	(E5)	221	DD	(15) 迷	253	FD	(M5) (1) 图
158	9E		190	BE	(E6)	222	DE	(16)	254	FE	(M6) (2)
159	9F		191	BF	(E7)	223	DF	(17) 路	255	FF	(M7) (3) 形

用 游 戏 机 作 曲

用游戏机作曲目录

一 音乐基本知识

二 计算机音乐程序的写法

利用游戏机（普及型电脑）的音乐功能，可以帮助我们学习作曲和为曲子配合声等。首先，让我们回忆一下音乐的基本知识（主要是简谱知识）。然后，看看该怎样把一首简谱记录的曲子输入到游戏机中，让游戏机来演奏。

一、音乐基本知识

乐曲都是由许多高、低、长、短、强、弱不同的音所组成的，把这些音记录下来就是乐谱。简谱是一种最常见的记录方法，它是用阿拉伯数字“1 2 3 4 5 6 7”来表示音乐中的 7 个基本音，并使用如下的一些符号来表示音的性质。

1. 音的高低：

首先介绍两个名词：

（1）音阶一乐音体系中的各音叫做音阶。

（2）音列一乐音体系中的音按照上行次序（即由低音到高音的排列次序）或下行次序排列，叫做音列。

在简谱中，使用阿拉伯数字“1 2 3 4 5 6 7”来表示 7 个基本音阶，7 个基本音阶，在其上方加点表示高音；下方加点表示低音，如下所示：

$\cdots\underset{\cdot}{1}\ \underset{\cdot}{2}\ \underset{\cdot}{3}\ \underset{\cdot}{4}\ \underset{\cdot}{5}\ \underset{\cdot}{6}\ \underset{\cdot}{7}\ \ 1\ 2\ 3\ 4\ 5\ 6\ 7\ \ \dot{1}\ \dot{2}\ \dot{3}\ \dot{4}\ \dot{5}\ \dot{6}\ \dot{7}\cdots$

低音部　　　　中音部　　　　高音部

整个音列范围被分成若干个音部，即音列中的一部分由“1”（do）到上面的“7”（si）就是一个音部。

2. 音　符

音符是个别乐音的标记，是乐曲中的最基本的单位。以 5 音为例：

全音符：	5 — — —	十六分音符：	$\underline{\underline{5}}$
二分音符：	5 —	三十二分音符：	$\underline{\underline{\underline{5}}}$
四分音符：	5	倚音音符：	$\frac{d}{\tau}5$
八分音符：	$\underline{5}$	震音音符：	$\tilde{5}$

音符有很多，不能一一例举。

3. 音的长短

我们知道了什么是音符，还应该知道怎样记录音符时值（即时间的长短）的长短，在简谱中有一个单纯音符“—”，用它来记录音的长短。它的用法有 2 种：

（1）写在音符的下面，表示这个音符的时值缩短为原来音符时值的一半，画两横就缩短为原来音符时值的四分之一，以此类推。

如：　5　四分音符

$\underline{5}$　八分音符

$\underline{\underline{5}}$　十六分音符

$\underline{\underline{\underline{5}}}$　三十二分音符

(2) 写在音符右边，表示音符时值增加一个四分音符时值的长度。

如： 5　　四分音符

5—　二分音符

5———全音符

4. **附点音符**

附点就是记在音符右边的小点，小点左边的音符叫符点音符。这个小点附在哪个音符的右边，哪个音符的时值便延长一半时间。

如：　　5·＝5＋5　　5·＝5＋5

5. **休止符**

在音乐中，有时需要停顿和休止，在简谱中用“O”来表示，这个“O”叫休止符。

6. **升降记号**

升记号＃：记在音符的前面，表示比基本音级高半音。如：＃5 叫做“升 5”。

降记号 b：记在音符前面，表示比基本音级降半音。如：b5 叫做“降 5”。

7. **节拍、拍子和小节**

节拍的作用是用来计算音的长短。

节拍的单位是用固定时值（二分音符，四分音符等）来代表的，它由“拍子”来规定。拍子用分数作标记，分子说明每小节内单位拍的数目，分母说明单位拍的时值。

例如：2/4 拍：二四拍，以四分音符为一拍，每小节有二拍。

3/4 拍：三四拍，以四分音符为一拍，每小节有三拍。

小节就是乐曲中的段落，相当于乐谱上所记载的由强拍开始到下一个强拍为止的部分。用垂线做为划分小节的小节线。

如：　| 5—5— | 6—6— |

有了上面所介绍的一些音乐基本知识，就知道一首乐曲是怎样用简谱记录的，并能够看懂它了。下面是大家所熟悉的《生日快乐歌》的简谱。

EX. 1　　　　生日快乐歌

C　3/4

| 5·5 6 5 | 1 7— | 5·5 6 5 | 2 1— |

| 5 · 5 5 3 | 1 7 6 | 4· 4 3 1 | 2 1— |

二、计算机音乐程序写法

1. 单部曲

在游戏机 FBASIC 语言中，为将简谱所记录的曲子输入到计算机中去，设计了一套新的代码。象简谱一样，它也能用来完整地表示一首乐曲，下面的表 1、表 2、表 3，就是这种代码与简谱的对应关系。

表 1　音域代码

音域代码	音　　阶
O0	1 2 3 4 5 6 7
O1	1 2 3 4 5 6 7
O2	1 2 3 4 5 6 7
O3	1 2 3 4 5 6 7
O4	1 2 3 4 5 6 7
O5	1 2 3 4 5 6 7

表 2　音阶代码

音阶	音阶代码
1	C
#1・b2	#C
2	D
#2・b3	#D
3	E
4	F
#4・b5	#F
5	G
#5・b6	#G
6	A
#6・b7	#A
7	B
O	R

表 3　音符长度代码

简谱记法	说　　明	音符长度代码
$\underline{\underline{\underline{0}}}$	三十二分音符	0
$\underline{\underline{0}}$	十六分音符	1
$\underline{\underline{0}}$・	附点十六分单符	2
$\underline{0}$	八分音符	3
$\underline{0}$・	附点八分音符	4
0	四分音符	5
0・	附点四分音符	6
0—	二分音符	7
0— 0—	附点二分音符	8
0···	全音符	9

表 1 是音域代码，将整个音列分成 6 个音部，分别用 O0 到 O5 表示。

表 2 是音阶代码，用字母及升记号“#”和降记号“b”来表示音阶。

表 3 是音符长度代码，用数字 0—9 来表示音的长度。

〈规则 1〉对于每一个音符，用表 1、表 2、表 3 的代码一起来表示，顺序是：音域代码、音阶代码、音符长度代码。

如：简谱表示　计算机音乐代码表示

$\underline{5}$・	O3G4	O3 表示其后的音阶 G 在中音区。 G 是 5 的音阶代码。 4 表示音符长度为附点的 1/8 音符。
$\underline{\underline{5}}$	O3G1	
6	O3A5	
5	O3G5	

〈规则 2〉如果后面的音符与前面的音符在同一音部中，则音域代码可省略。象上面四个音符的例子把它们连起来｜ $\underline{5}$・ $\underline{\underline{5}}$ 6 5 ｜用计算机音乐代码表示为。

｜O3　G4　G1　A5　G5｜

下面是《生日快乐歌》的简谱和它的计算机乐谱。读者可对照看：

EX. 1　　　　生日快乐歌

C 3/4

| 5·565 | 17 — | 5·565 | 21 — |

| 5·553 | 176 | 4·431 | 21 — |

‖ O3G4G1A5G5 | O4C5O3B7 | G4G1A5G5 | O4D5C7 |

| O3G4G1O4G5E5 | C5O3B5A5 | O4F4F1E5C5 | D5C7 ‖

〈规则 3〉如果后面音符与前面音符的音符长度相同，则后面音符的音符长度代码可以省去。上面的计算机乐谱又可简化为如下形式：

‖ O3G4G1A5G | O4C5O3B7 | G4G1A5G | O4D5C7 |

| O3G4G1O4G5E | C5O3BA | O4F4F1E5C | D5C7 ‖

〈规则 4〉在 BASIC 语言中表示音乐的指令是 PLAY""，现在可以把上面计算机乐谱所表示的曲子写成 BASIC 语言的形式，输入到计算机中，并试听演奏效果了。

```
20 PLAY"O3G4G1A5G"
30 PLAY"O4C5O3B7"
40 PLAY"G4G1A5G"
50 PLAY"O4D5C7"
60 PLAY"O3G4G1O4G5E"
70 PLAY"C5O3BA"
80 PLAY"O4F4F1E5C"
90 PLAY"D5C7"
100 END
```

在上面的程序中，我们没有指定速度、音调、音量和效果音，演奏出的效果是计算机中初始设定的速度、音调、音量及音色效果。下面的表 4、表 5、表 6 给出这些代码。

表 4 速度代码

速度	代码
♩=400	T1
♩=220	T2
♩=150	T3
♩=110	T4
♩=90	T5
♩=80	T6
♩=65	T7
♩=60	T8

表 5 音量代码

代码	说明
M1Vn	M1 表示顿音效果，V1—V4 时有明显顿音效果，V5—V15 时则只有短音效果，音量固定（=V15）
M0Vn	M0 时为一般情形，Vn 表示音量大小，V1（小）↔V15（大）

表 6 音色效果

音色效果	代码
12.5%	Y0
25.0%	Y1
50.0%	Y2
75.0%	Y3

表 4 是速度代码表，用 T1－T8 来表示演奏的速度，T1 最快，T8 最慢。

表 5 是音量代码表。

表 6 是音色效果的代码表。其中的百分数是指所输出方波的占空比。

根据表 4—表 6，我们把上面《生日快乐歌》的程序加上第 10 语句，就可得到如下的完整的

音乐程序：

```
10  PLAY"M1V10Y2T4"
20  PLAY"O3G4G1A5G"
30  PLAY"O4C5O3B7"
40  PLAY"G4G1A5G"
50  PLAY"O4D5C7"
60  PLAY"O3G4G1O4G5E"
70  PLAY"C5O3BA"
80  PLAY"O4F4F1E5G"
90  PLAY"D5C7"
100 END
```

如果希望连续演奏上面的乐曲，可以将100　END语句改为100　RUN，此时，计算机将反复演奏，直到在键盘上按Pause键为止。

M1Vn适合轻快的乐曲；M0Vn适合舒情歌曲。通过反复的使用和理解，就会掌握自己所喜欢的音色了。

下面是一首练习曲《小毛驴》

EX. 2　　　　　　　　　　　　　　小　毛　驴

F　2/4

‖ 1 1 1 · 3 | 5 5 5 · 5 | 6 6 6 i | 5 · 0 | 4 4 4 6 | 3 3 3 3 | 2 2 2 2 | 5 · 5 |

| 1 1 1 · 3 | 5 5 5 · 5 | 6 6 6 i | 5 · 0 | 4 4 4 6 | 3 3 3 3 3 3 | 2 2 2 3 | 1 — ‖

```
10  PLAY"M1V14Y3T3"
20  PLAY"O2C3CC4E1"
30  PLAY"G3GG4G1"
40  PLAY"A3AAO3C"
50  PLAY"O2G6R3"
60  PLAY"F3FFA"
70  PLAY"E3EEE"
80  PLAY"D3DDD"
90  PLAY"G6G3"
100 PLAY"C3CC4E1"
110 PLAY"G3GG4G1"
120 PLAY"A3AAO3C"
130 PLAY"O2G6R3"
140 PLAY"F3FFA"
150 PLAY"E1EEEE3E"
160 PLAY"D3DDE"
170 PLAY"C7"
180 END
```

练习 3 和练习 4 给出两首乐曲的简谱，请读者写出它们的计算机音乐程序，输入计算机试奏出来。

EX. 3　　　　梅　花

C　3/4

‖: 5 — 3 | 6 — 3 | 2 0 3 1 6 | 5 — — | 6 1 6 | 5 — 6 | 3 — — | 3 — — |

| 5 — 3 | 6 — 3 | 2 0 3 1 | 6 — — | 5 6 5 | 3 — 2 | 1 — — | 1 — — :‖

EX. 4　　　　欢　乐　颂

G　4/4

‖ 3 3 4 5 | 5 4 3 2 | 1 1 2 3 | 3 · 2 2 0 |

| 3 3 4 5 | 5 4 3 2 | 1 1 2 3 | 2 · 1 1 0 |

| 2 2 3 1 | 2 34 3 1 | 2 34 3 2 | 1 2 5 3 |

| 3 3 4 5 | 5 4 3 2 | 1 1 2 3 | 2 · 1 1 0 ‖

2、二部曲

〈规则 5〉练习 5 是一首二部曲，二部曲的计算机音乐程序的表达方法是：以小节为单位，将两声部的音乐代码分开写，中间用冒号隔开。

EX. 5　　　　小　乖　乖

C　4/4

‖: 1 2 3 4 5 6 5 | 4 2 7 5 3 1 | 1 2 3 4 5 6 5 | 0 0 1 1 1 |

| 4 2 7 2 1 — |

| 1 — — 1 1 | 3 — — 7 6 | 5 — — 7 6 | 5 — — 1 1 | 1 — — 1 1 |

| 6 6 1 7 6 | 5 3 5 — | 4 2 7 — | 5 3 1 — | 6 6 1 7 6 |

| 3 — — 3 3 | 4 — — 4 | 3 — — 0 ‖

| 5 3 5 — | 4 2 7 2 | 1 — — 0 ‖

```
10 PLAY"M0V10Y3T4：M0V10Y2T4"
20 PLAY"O2C3DEFGAG5"
30 PLAY"F3DB5G3EO3C5"
40 PLAY"O2C3DEFGAG5"
50 PLAY"F3DBO3DC7：R7O3C5C3C"
60 PLAY"O2A5AO3CO2B3A：C8C3C"
```

```
70  PLAY"G5EG7：E8O2B3A"
80  PLAY"F5DB7：G8B3A"
90  PLAY"G5EO3C7：G8O3C3C"
100 PLAY"O2A5AO3CO2B3A：C8C3C"
110 PLAY"G5EG7：E8E3E"
120 PLAY"F5DBO3D：F8F5"
130 PLAY"C8R5：E8R5"
140 END
```

EX. 6　　　　　　　　我 的 一 毛 钱

```
3 2 ‖ 1 1 1 3 4 | 5 5 5 1 2 | 3 3 3 1 2 | 3 2 2 3 2 |
    ‖ 1 3 5 —   | 7 7 7 1 7 | 1 1 1 1 7 | 1 7 7 0   |

| 1 1 1 3 4 | 5 5 5 0 | 3 5 6 5 3 1 2 | 3 3 2 2 1 0 |
| 1 3 5 —   | 7 7 7 0 | 1 3 4 3 1 1 7 | 1 1 7 7 5 0 |

| 3 5 6 5 3 1 2 | 3 3 2 2 1 0 ‖
| 1 3 4 3 1 1 7 | 1 1 7 7 5 0 ‖
```

```
00  REM"MY  ONE CENT"
10  PLAY"M1V10Y3T3：M1V10Y2T3"
15  PLAY"O3E3D"
20  PLAY"C5CCO2E3F：O3CEG7"
30  PLAY"G5GGO3C3D：O2B5BBO3C3O2B"
40  PLAY"E5EEC3D：O3C5CCC3O2B"
45  PLAY"E5DDE3：O3C5O2B5BR"
50  PLAY"C5CCO2E3F：O3C5G7"
60  PLAY"G5GGR：O2B5BBR"
70  PLAY"03E3G5A3GECD：O3C3E5F3ECCO2B"
80  PLAY"EEDDC5R：O3CCO2BBG5R"
90  PLAY"E3G5A3GECD：O3C3E5F3ECCO2B"
100 PLAY"EEDDC5R：O3CCO2BBG5R"
110 END
```

〈规则 6〉在 BASIC 语言的音乐语句中，引号中的字符不能超过 31 个，如果程序太长，必须分段输入。

下面给出一首二部曲的乐谱，请读者自己试写出计算机音乐程序。

EX. 7 野　餐

F 4/4

D 5 · 5 6 5 3 1 | G 1 · 6 6 0 | D 5 · 1 3 1 5 3 | A7 2 — — 0 |
1 · 1 1 1 1 5 | 6 · 4 4 0 | 5 · 5 5 5 1 1 | 7 — — 0 |

D 5 · 5 6 5 3 1 | G 1 · 6 6 0 | D 5 · 1 3 2 1 7 | D 1 — — 0 |
1 · 1 1 1 1 5 | 6 · 4 4 0 | 3 · 5 1 7 6 5 | 5 — — 0 |

EM# 2 · 1 2 3 4 2 | D 3 — 5 0 | G 6 · 6 5 3 4 3 | A7 2 — — 0 |
7 · # 6 7 1 2 7 | 1 — 3 0 | 1 · 1 1 1 2 1 | 7 — — 0 |

D 5 · 5 6 5 3 1 | C 1 · 6 6 0 | D 5 · A7 1 3 2 1 7 | D 1 — — 0 ‖
1 · 1 1 1 1 5 | 6 · 4 4 0 | 3 · 5 1 5 5 5 | 5 — — 0 ‖

3、三部曲

〈规则 7〉三部曲的语法规则与二部曲一样，只是第三部低音部的音量（V）和音色效果（Y）是固定的，由计算机指定，读者不必输入音量代码和音色效果代码。

EX. 8 望　春　风

A 4/4

5 · 5 6 1 | 2 3 2 1 2 3 — | 5 · 3 3 2 1 | 2 — — — |
3 · 3 4 6 | 7 3 1 — | 3 · 1 1 1 | 7 — 5 6 1 2 |
1 — 4 — | 5 — 6 — | 7 — 6 — | 5 — — — |

3 · 5 5 3 5 | 1 · 2 2 — | 5 · 3 3 2 | 1 — — — |
5 · 3 3 5 | 6 · 7 7 — | 2 · 1 1 7 | 3 — 1 7 6 5 |
1 — 5 — | 4 — 5 — | 4 — 5 — | 1 — — — |

2 · 2 3 2 1 | 6 5 6 1 — | 6 · 1 2 3 | 5 — — — |
7 · 7 1 7 | 1 2 1 6 5 | 1 · 6 7 1 | 3 · 5 5 4 3 |
5 — — — | 4 — — — | 3 — — — | 2 — — — |

5 · 5 6 5 3 | 3 2 1 6 — | 5 · 3 3 2 | 1 — — — ‖
3 · 3 1 3 1 | 5 5 1 — | 7 · 1 1 7 | 3 — — — ‖
1 — 4 5 | 1 3 4 — | 5 · 5 — | 1 — — — ‖

```
101 PEM"WANG CHUEN FENG"
105 FOR N=1 TO 2
107 PLAY"M0V14Y3T4；M0V10Y2T4；M0T4"
```

```
110 PLAY"O2G6G3A5O3C：O2E6E3F5A：O1C7F"
120 PLAY"D4E1DC3DE7：B5O3EC7：GA"
130 PLAY"G6E3EDC5：E6C3C5C：BA"
140 PLAY"D9：O2B7G3AO3CD：G9"
150 PLAY"E6G3G5E3G：G6E3E5G：C7G"
160 PLAY"C6D3D7：O2A6B3B7：FG"
170 PLAY：O2G6O3E3E5D：O3D6C3C5O2B：FG"
180 PLAY"C9：O3E7C3O2BAG：C9"
190 PLAY"D6D3E5D3C：B6B3O3C5O2B：G9"
200 PLAY"O2A5G3AO3C7：O3C5DCO2A3G：F"
210 PLAY"O2A6O3C3D5E：O3C6O2A3B5O3C：E"
220 PLAY"G9：E6G3G5F3E：D"
230 PLAY"G6G3A5G3E：E6E3C5E3C：C7F5G"
240 PLAY"E5D3CO2A7：G5GC7：C5EF7"
250 PLAY"G6O3E3E5D：O2B6O3C3C5O2B：GG"
260 PLAY"C9：O3E9：C9"
270 NEXT
280 END
```

下面给出一首三部曲的曲谱，请读者自己试写出计算机音乐程序。

EX. 9　　　　余　烬

C4/4

```
                                 G7        C
‖ 5 · 4 3 5 1 2 | 3 — 1 — | 2 1 7 2 1 6 | 5 — — 0 |
‖ 3 · 2 1 3 3 4 | 5 — 3 — | 4 4 4 4 3 4 | 3 — — 0 |
‖ 1 · 1 1 1 1 5 | 1 — 1 — | 5 5 5 5 1 1 | 1 — — 0 |

  C                 Dm        G7              C
| 5· 4 3 5 1 2 3 | 2 — 6 — | 7 · 6 5 7 1 2 | 1 — — 0 ‖
| 3 · 2 1 3 3 5  | 4 — 4 — | 4 · 4 4 4 3 4 | 3 — — 0 ‖
| 1 · 1 1 1 1 5  | 6 — 2 — | 5 · 5 5 5 5 7 | 1 — — 0 ‖
```

4. 乐曲例子

EX. 10　　　　往 事 难 忘

F　4/4

```
‖ 1 1 2 3 3 4 | 5 · 5 6 5 3 — | 5 4 3 2 —     | 4 3 2 1 — |
| 1 1 2 3 3 4 | 5 · 5 6 5 3 — | 5 4 3 2 3 2   | 1 — — —   |
| 5 4 3 2 5 5 | 4 3 2 1 —     | 5 4 3 2 5 5   | 4 3 2 1 — |
| 1 1 2 3 3 4 | 5 6 5 3 —     | 5 4 3 2 3 2   | 1 — — —   ‖
```

```
00  REM"LONG LONG AGO"
10  PLAY"MIV10Y0T3: M1V14Y2T3"
15  FOR X=1 TO 2
20  PLAY"O3C3GCDEGEF: O3C5C3DE5E3F"
30  PLAY"GO4CO3AGEGO4CO3G: G5A3GE7"
40  PLAY"GBFEDGBG: G5F3ED7"
50  PLAY"FGEDCEGE: F5E3DC7"
60  PLAY"CGCDEGEF: C5C3DE5E3F"
70  PLAY"GO4CO3AGEGO4CO3G: G5A3E7"
80  PLAY"GBFEDGED: G5F3ED5E3D"
90  PLAY"CEGEC5R: C7C5R"
100 PLAY"G3BFEDGO2GO3G: G5F3ED5O2G"
110 PLAY"FGEDCEGE: O3F5E3DC7"
120 PLAY"GBFEDGO2GO3G: G5F3ED5O2G"
130 PLAY"FGEDCEGE: O3F5E3DC7"
140 PLAY"CGCDEGEF: C5C3DE5E3F"
150 PLAY"GO4CO3AGEGO4CO3G: G5A3GE7"
160 PLAY"GBFEDGED: G5F3ED5E3D"
170 PLAY"CEGEC7: C7C7"
180 NEXT
190 END
```

EX. 11　　　　　　　　给 爱 丽 丝

C 3/4

| 3#2 3#2 3 7 2 1 | 6 · 1 3 6 | 7 · 3 #5 7 | 1 · 3 3#2 |

| 3#2 3 7 2 1 | 6 · 1 3 6 | 7 · 3 1 7 | 6 — 3#2 |

| 3#2 3 7 2 1 | 6 · 1 3 6 | 7 · 3 #5 7 | 1 · 3 3#2 |

| 3#2 3 7 2 1 | 6 · 1 3 6 | 7 · 3 1 7 | 6 · 7 1 2 |

| 3 · 5 4 3 | 2 · 4 3 2 | 1 · 3 2 1 | 7 — 3 3 |

| 3 — 3 #2 | 3 — 3 #2 | 3#2 3 7 2 1 | 6 · 1 3 6 |

| 7 · 3 #5 7 | 1 · 3 3 #2 | 3#2 3 7 2 1 | 6 · 1 3 6 |

| 7 · 3 1 7 | ① 6 · 7 1 2 :‖ ② 6 — ‖

```
00  REM"FOR  ELISE"
```

```
10  PLAY"MIV10Y2T8: M1V10Y2T8"
20  PLAY"O4E1#DE#DEO3BO4DC: R7"
30  PLAY"O3A3R1CEA: O2R1EAR1R3"
40  PLAY"B3R1E#GB: R1EBR1R3"
50  PLAY"O4C3R1O3EO4E#D: R1EAR1R3"
60  PLAY"E#DEO3BO4DC: R5R3"
70  PLAY"O3A3R1O3CEA: R1EAR1R3"
80  PLAY"B3R1EO4CO3B: R1EBR1R3"
90  PLAY"A3R1R1O4E#D: R1EAR1R3"
100 PLAY"E#DEO3BO4DC: R5R3"
110 PLAY"O3A3R1CEA: R1EAR1R3"
120 PLAY"B3R1E#GB: R1EBR1R3"
130 PLAY"O4C3R1O3EO4E#D: R1EAR1R3"
140 PLAY"E#DEO3BO4DC: R5R3"
150 PLAY"O3A3R1CEA: R1EAR1R3"
160 PLAY"B3R1EO4CO3B: R1EBR1R3"
170 PLAY"A3R1R3: R1EABO3CD"
180 PLAY"E1O4EER1R3: R1R3O2G1O3FE"
190 PLAY"O3D1O4FFR1R3: R1R3O2F1O3ED"
200 PLAY"O3C1O4EER1R3: R1R3O2E1O3DC"
210 PLAY"O2B1O3EO4ER1O3EE: O2B3R1O2E4"
220 PLAY"O1ER1O3E#GEO4#D: R1ER5"
230 PLAY"EO3#DEO4#DE#D: R3R5"
240 PLAY"E#DEO3BO4DC: R3R5"
250 PLAY"O3A3R1CEA: R1EAR1R3"
260 PLAY"B3R1E#GB: R1EBR1R3"
270 PLAY"O4C3R1O3EO4E#D: R1EAR1R3"
280 PLAY"E#DEO3BO4DC: R3R5"
290 PLAY"O3A3R1CEA: R1EAR1R3"
300 PLAY"B3R1EO4CO3B: R1EBR1R3"
305 IF X=1  THEN  330
310 PLAY"A3R1R1R3: R1EABO3CD"
320 X=1: GOTO  180
330 PLAY"A8"
340 END
```

EX. 12

楚留香新传

Em 4/4

Em 6 · 1 3 5 | 6 — — 1 6 | C 5 6 5 3 1 | Am 2 3 2 CD 1 7 | Em 6 — — — | 6 — — — |

6 — — 1 7 ‖: Em 6 · 1 3 5 | 6 — — 1 6 | G 5 · 6 5 3 · 5 |

Am 2 — — — | Am 2 · 3 6 1 | Bm 5 · 6 5 3 — | Am 2 2 1 C 7 D 7 5 |

Em 6 — — 1 7 :| Am 2 2 1 C 1 D 7 5 | Em 6 — — 1 7 | 6 · 1 3 6 |

G 5 — — 6 5 | 3 · 5 6 3 | Am 2 — — 3 2 | Em 1 · 6 6 6 | Bm 5 — 3 3 3 |

C 2 2 2 1 D 7 7 7 5 | Em 6 — — 1 7 | Em 6 1 3 6 | G 5 — — 6 5 |

Em 3 · 5 6 3 | Am 2 — — 3 2 | Em 1 · 6 6 6 | Bm 5 — — 3 C 5 | 6 2 D 1 · 7 |

Em 6 — — 1 7 ‖ Bm 5 — — 3 5 | C 6 2 1 · 7 | Em 6 — — — |

6 — — 1 6 | 5 3 6 5 3 1 | Am 2 3 2 C 1 D 7 | Em 6 — — — | 6 — — — ‖

Rit……………………………………Fine

```
10 PLAY"M0V14Y3T4: M0V14Y1T4: M1T4"
20 PLAY"O3A6O4C3O3E5G"
30 PLAY"A8O4C3O3A"
40 PLAY"G5A3GE5C"
50 PLAY"D5E3DC5O2B"
60 PLAY"A7R8: O2R7E5GAB8"
70 PLAY"O4C3O3B"
74 Y=0
75 FOR X=1 TO 2
80 PLAY"A6O4C3O3E5G: O3A9: O1A3AAAAAAA"
90 PLAY"A8O4C3O3A: C7E: A3AAAAGFE"
100 PLAY"G6A1GE6G3: CG: GGGGGGGG"
110 PLAY"D9: F5EDC: DDDDDDDD"
120 PLAY"D6E3O2A5O3C: G7F: GGGGGFED"
130 PLAY"G6A1GE7: E7G: O1EEEEEEEE"
140 PLAY"D3D5C3O2B5B3G: DF: GGGGGGGG"
150 PLAY"A8O4C3O3B: C1DEFEFGAB7: AAAAAAAA"
160 NEXT
170 PLAY"A6O4C3O3E5A: A6A3A5E: A3AAAAABA"
180 PLAY"G8A3G: D6G3G5F: GGGGGGAG"
190 PLAY"E6G3O2A5O3E: C6G3C5C: AAAAAAAA"
```

```
200 PLAY"D8E3D; F6F3D5D; DDDDDDDD"
210 PLAY"C6O2A3A5O3A; E6E3C5E; AAAAAAAA"
220 PLAY"G7E5E3E; E6E3G5G; O2CCCCCCCC"
230 PLAY"DDDCO2BBBG; F6F3G6D3; DDDDGGGG"
240 PLAY"A8O4C3OB; O6E3E5C; O1AAAAAAAA"
250 PLAY"A6O4C3OE5A; C6E3A5E; AAAAAAAA"
260 PLAY"G8A3G; B6B3B6D3; GGGGGGGG"
270 PLAY"E6G3O2A5O3E; C6C3C5C; AAAAAAAA"
280 PLAY"D8E3D; F6D3D5G; DDDDDDDD"
290 PLAY"C6O2A3A5O3A; E6C3C5C; AAAAAAAA"
300 PLAY"G8E3G; E6E3G5G; O2CCCCCCCC"
305 IF Y=1 THEN  340
310 PLAY"A5O4DC6O3B3; C5FEE; O1AAO2DDCCCC"
320 PLAY"A8O4C3O3B; C9; O1AAAAAAA"
330 Y=1; GOTO  75
340 PLAY"A5O4D; C5F; O1AAO2DD"
345 PLAY"C7O3B7· E7E; C7O1B"
350 PLAY"A9; A9; A9"
360  END
```

EX. 13　　　　　明 天 会 更 好

B—C　4/4

C　Em　F　Em　Dm7

0 3 4 ‖ 5 — — 0 3 2 | 1 — — 0 3 4 | 5 — 0 1 5 3 4 |

G7　C　Em　G7

4 — — 0 3 4 | 5 — 0 1 1 · 7 | 6 — — 0 3 4 | 5 — 0 · 4 3 · 2 |

C　Em　F　C　G

1 — — 0 3 4 ‖: 5 5 5 5 5 6 5 4 4 3 4 | 5 5 3 2 1 2 · 3 2 |

Am　G　F　G　Am　G

1 1 1 1 1 · 7 1 7 5 5 | 6 5 4 4 4 5 6 5 — | 1 1 6 3 5 6 5 5 |

F　G　C　Am　I　Dm

6 6 5 1 2 3 · 2 ‖ 1 · 1 1 3 3 3 2 2 | 2/4 2 2 7 6 ‖

G　Am　III II　Dm　G

‖ 4/4 6 5 5 — 0 3 4 :‖ 1 · 1 1 3 3 3 2 6 | 2/4 5 · 3 2 5 3 2 ‖

C　Am　Em　F　G　C

| 4/4 1 — · — | 6 6 6 7 1 6 7 7 7 1 2 7 | 1 7 6 5 5 5 1 2 3 — |

Bm F Am E7

| 3 77733 2 2 1 7 6 | 6 — · — | 666 7 1 6 · 7 7 7 2 7 7 |

Am F G C C

| 3 · 2 1 · 7 | 6 6 66 · 5 5 2 3 · | 1 — — 0 3 4 ‖ ‖ 1 — · — |

C# C# G#

| 0 0 0 0 3 4 ‖ 55 5 5 5 6 5 4 4 34 | 5 5 3 2 1 2 · 3 2 |

A#m G# F# C# A#m

| 111 1 1 · 7 1775 | 6 5 4 4 4 5 6 5 — | 1 1 6 3 5 655 |

F# C# C# A#m DAm C#7

| 6 6 5 1 2 3 · 2 ‖ 1 · 1 1 3 3 3 2 6 ‖ 2/4 5 · 3 2 5 2 3 |

C# A#m Fm F# C#

| 4/4 1 — · — ‖ : 6.6 6 7 1 6 7 7 7 1 2 7 | 1 7 6 5 5 5 1 2 3 — |

Fm F# A#7 F7

| 37 7 7 3 3 2 2 1 76 | 6 — · — | 6 6 6 7 1 6 · 7 7 7 2 7 7 |

A#m F# G#7 C#

| 3 · 2 1 · 7 | 6 6 6 6 · 5 5 2 3 · | 1 — · — : ‖

```
00 REM"TOMORROW WILL BE BETTER"
03 CLS
04 PALETS 0, 7, 18, 18, 18
05 LOCATE 2, 9: PRINT"*TOMORROW WILL BE BETTER*"
10 PLAY"M0V10Y1T5: M0V14Y3T5: M0T5"
20 PLAY"O3E1FG8R3E1DC8R3E1F"
30 PLAY"G7R3O4CO3GE1F: O1C9"
40 PLAY"F8R3E1F: F9"
50 PLAY"G7R3CO4C4O3B1: G"
60 PLAY"A8R3E1F: A"
70 PLAY"G7R4F1E4D1: G"
80 PLAY"O3C3GO4EO3GO4EO3G: C8: C8"
90 PLAY"O4EO3G: R3E1F: R5"
102 PALETS 0, 12, 30, 30, 30
106 X=0: Y=0: Z=0: W=0
110 PLAY"O4E3O3GO4EO3G: G3GGG1G: C5E"
112 PLAY"O4CFDO3A: AGF3FE1F"
120 PLAY"E3O3GO4EO3G: G3GE1DC3: C7"
125 PLAY"BGO4C1DEF: D6E1D: G"
130 PLAY"G3CEC: C1CC3O4C1C4: O2C"
135 PLAY"FGEG: O3B1O4CO3BG5R1: O1B"
```

```
140 PLAY"CFO3AO4F：A1GFF3G1A3：A"
145 PLAY"DGDG：G7：G"
150 PLAY"O3G3O4EO3GO4E：O4C3C5O3A1E：O2C"
155 PLAY"O3GO4EDO3B：GAG6：O1B"
160 PLAY"O4E3CEO3A：A3AG1C3D1：A"
162 PLAY"O4EDCO3B：E6D3：G"
165 IF X=1 THEN 200
170 PLAY"O4C3O3AO4CO3A：O4C1O3E1E：F"
175 PLAY"O4GFGD：E3D5D3：E"
180 PLAY"F3O3AO4F：O4D3DO3B：D6"
185 PLAY"O3B1O4CDEFG7R5：A5G7R5E1F：D3C9"
190 X=1：GOTO110
200 PLAY"O3A3O4FO3AO4F：C4C1C3E1E：F7"
205 PLAY"O3GBGB：E3D5A3：F"
210 PLAY"O4F3O3GBG：G4E1DO2GO3ED：D"
215 PLAY"O4A3GEDCDC：C9：C9"
216 PALETS 0，10，25，25，25
220 PLAY"O3A1BO4CEO3AB：A1AABO4C3：A6"
225 PLAY"O4CEO3ABO4CE：O3A1B3BO4C1：A0A2G5"
226  PLAY"GGGG：D3O3B：G3E"
230  PLAY"CO3BAGO4GG：O4C1O3BAG3G1：F6"
232  PLAY"GG：CD：G3"
235  PLAY"EEEEECDC：E7：C6G3"
240  PLAY"GBGBGBGB：E3B1B4E1E：E6R0E2"
245  PLAY"O5DCO4BA：O4D3D：E5"
247  PLAY"O5CO4BA#G：C1O3B3A1：E3G"
250  PLAY"AAAAACEGACEGCO3BAG：A9：A5GFE"
260  PLAY"O3A1BO4CEAGFE：A1AABO4CO3A4：A6G3"
265  PLAY"O3ABO4CEGG：B1B3O4D3：E6"
267  PLAY"GG：O3B1B：F3"
270  PLAY"O5DCO4B#G：O4E5：O2E5"
272  PLAY"O5DCO4B#G：R3D：E3D"
275  PLAY"AO5CDEO4AO5CDE：C6O3B3：C5O1B3"
280  PLAY"CCCCCCCC：A3AA1A4：A6R0A2"
285  PLAY"O4BBBBGGGG：G3GD1E4：G6R0G2"
287  IF W=1 THEN 340
288  IF Z=1 THEN 311
289  IF Y=1 THEN 311
290  PLAY"O5C8R5：C8R3E1F：C8R5"
299  PALETS 0，4，40，25，10
```

```
300  Y=1: GOTO 110
311  PLAY"O4C7: O3C7: C7"
314  IF Z=1 THEN 330
320  Z=1: GOTO 220
330  W=1: GOTO 220
340  PLAY"O4C7: O3C7: C7"
350  PLAY"C1CCCCCCC: A3AA1A4: A6R0A2"
360  PLAY"O4BBBB: O4D3D: MID3D"
365  PLAY"GGGG: DIE4: DIM0E4"
370  PLAY"O5C9: C9: C9"
380  PALETS 0, 8, 39, 39, 39
385  CLS
390  LOCATE 11, 9: PRINT"*END*"
400  LOCATE 9, 10: PRINT"THANK  YOU!"
410  END
```

LOGO 语言入门手册

前　言

LOGO 语言是美国麻省理工学院人工智能实验室专为青少年和初学者设计的计算机语言。它不同于传统的计算机语言，它以孩子们十分喜欢的搭积木拼图的方式，通过海龟在屏幕上绘图来学习编制程序，训练学生的逻辑思维和创造才能。目前，LOGO 语言在国外已十分流行，在我国也已成为中小学计算机教学的内容之一，深受广大青少年的喜爱。

LOGO 语言的主要特点是：简单易学；程序具有模块化结构，每个程序可以由一个或几个相对独立的模块(过程)组成，设计方便灵活；LOGO 的模块一经定义便成了一个新的命令，以后可以用来作为原始指令调用，因此命令很容易扩充；LOGO 具有 BASIC 不具备的递归功能（自己调用自己），可编制高水平的应用程序；LOGO 语言不仅包括数字和字符串，还包括一种能将数字和字符串自由排列的表结构，处理数据方便有效。

本手册介绍的所有操作都已在裕兴超级学习卡所配的 LOGO 语言状态下通过。如您能愉快地依次读下去，想来必能循序渐进，学有所成。如果你在阅读中有不明了的地方，不妨向兄长、父母或是老师请教一下。

LOGO 语言入门手册目录

一　进入 LOGO 世界

1.1　进入 LOGO 世界

要想进入 LOGO 世界首先要明确您所采用的超级学习卡的版本，4.0、4.0A、4.0C 版本按 F4 键进入，5.0 版本的按 F5 键进入多功能菜单，再按数字键 6 进入 LOGO 语言状态。5.0 以下的版本，要求在扩展插座上插有 LOGO 语言芯片。经上述确认后，在学习机上插好超级学习卡，开机后按指定键便可进入 LOGO 语言状态。这时，屏幕被清理干净，在屏首将显示 LOGO 的版本号和制造商，请注意 LOGO 语言的版本号需同卡的版本号一致。这时的屏幕方式正处于纯文本屏幕。

LOGO 的屏幕格式有三种：

（1）纯文本屏

它是用来显示程序及程序执行所得的文字结果的，共有 25 行，每行能显示 28 个字符。它主要用于人机对话，不能显示图形。

（2）纯图形屏

它是用来显示程序及程序执行所得的图形结果的，是体现 LOGO 优秀绘画能力的重要窗口，宽 256 点，高 240 点，每个字符由 8×8 个点组成。

（3）图文混合屏

这是 LOGO 使用对话式绘图时使用的屏幕。在这种状态下，在屏幕的底部会出现五行文本，用于显示程序及程序执行所得的文字结果，上部依旧显示图形，但是这五行文本会遮盖下部的图形。要想看到完整的图形请切换到纯图形文本屏，要想看到完整的程序请切换到纯文本屏。

三种屏幕之间可以任意切换，切换的方法是：按 F1 键或键入 TEXTSCREEN 命令，进入纯文本屏；按 F2 键或键入 SPLITSCREEN 命令，进入图文混合屏；按 F3 键或键入 FULLSCREEN 命令，进入纯图形屏。我们来试试吧！

进入图文混合屏或纯图形屏时，在屏幕正中间出现一个尖头朝上的白色三角形。啊哈！它就是我们的主角，大家都叫它“海龟”。它实际上相当于一支画笔，屏幕有如一张白纸。我们可通过键入命令或执行程序来控制海龟的位置、移动、方向、画图和着色，在屏幕上描绘出丰富多彩的图形。

1.2　走吧！小海龟！

我们先进入最常用的图文混合屏。海龟静静地趴在屏幕的中央，我们怎么能让它动起来呢？

请键入 FD 50，然后按回车键。这时，你会看到海龟向前爬了一段距离，身后留下一条笔直的轨迹。它动啦！再试试 BK 30，然后按回车键，它又退回一段距离，真有意思。你还可以命令它走不同的步数，海龟的一步相当于屏幕上的一个点。

如 FD 90

　BK 70

海龟不仅会画直线，还会拐弯呢！

如 LT 30，海龟就在当前位置逆时针转过 30 度，如 RT 95，它又顺时针转过 95 度，再 FD 20，你会看到它在屏幕上斜着前进了。现在你已经学会控制海龟向前进或后退了。

例　正方形

```
FD  50  RT  90
FD  50  RT  90
```

FD　50　RT　90

FD　50　RT　90

例　正五边形

FD　30　RT　72

FD　30　RT　72

FD　30　RT　72

FD　30　RT　72

FD　30　RT　72

不一会儿，屏幕上已经画得乱七八糟了。也许你想把它们全擦掉，重新作图，那么要用 CS 命令；

如果你想叫海龟回到屏幕中央但又不清屏，用 HOME 命令；如果既要海龟回到屏幕中央又要清屏，用我们上面提到的屏幕切换法就行，或者用 DRAW 命令；如果想退出作图状态，并清除所有的图形和文字，要用 ND 命令；如果你处在纯文本屏幕时，想清除屏幕上的文字，可用 CLEAR-TEXT 命令。

LOGO 命令是由英文字母组成的，有的命令后还要带上一个数或变量，这种命令后面所带的数或变量，我们叫它参数。比如 FD 等后面要带一个数或变量，如 FD 50，**注意命令与参数之间必须用空格隔开，这一点要时刻注意。**命令和参数组成语句，语句是 LOGO 执行的最小单位，多个语句可写在同一行内，**语句之间用空格隔开**，比如 FD 50 BK 30，这样的行称为语句行，一个语句行最多不能超过 255 个字符（包括空格），以回车结束一个语句行。但我们不提倡一行多句。如果键入的命令不正确，LOGO 会显示错误信息，有关错误信息请查阅附录。**再次提醒你一定留神空格!!!**

好了，前面已经学了很多知识，让我们来总结一下：

命令	功能
FD :N	前进 :N 步
BK :N	后退 :N 步
RT :M	右转 :M 度
LT :M	左转 :M 度
HOME	海龟回到母位，但不清屏
CS	清屏，但海龟不回到母位
DRAW	进入作图，海龟回到母位，清屏
ND	退出作图，图文全清
TEXTSCREEN	或 F1 进入纯文本屏
CLEARTEXT	清除纯文本屏
SPLITSCREEN	或 F2 进入图文混合屏
FULLSCREEN	或 F3 进入纯图形屏

命令中的带冒号的如 :N 或 :M 叫变量，我们将在第三章中详细解释，现在只需用具体数字取代它们就行了。

在纯文本状态下，可键入 GOODBYE 删除一切，返回初始状态。要想返回中西文浮点 BASIC 状态，得按复位键或是电源开关。

1.3 四海为家

屏幕正中间称为母位。图形屏上的每一点都要有一个名字，我们才能知道海龟处在什么位置。我们知道表示平面上的一个点至少要用一对数，就是我们常听说的横坐标和纵坐标。LOGO 语言规定图形屏幕上的每一点用相对于母位的横坐标和纵坐标来表示，母位的坐标为（0，0），向上 Y 轴最大的坐标为＋120，向下 Y 轴最小的坐标为－119，向右有 X 轴最大的坐标为的＋127，向左有 X 轴最小的坐标为－128，如图所示：

```
         +120
          ↑
-128← △ →+127
          ↓
         -119
```

比如有一点 A 坐标为（－60，－40），我们也可说成左 60 下 40。海龟不仅能前进、后退和拐弯，还能直接到达屏幕上的任意一点。请试以下命令：

SETX :X　　海龟移到横坐标为 :X 的位置

例 SETX 0

SETY :Y　　海龟移到纵坐标为 :Y 的位置

例 SETY　100

SETXY :X :Y　海龟移到坐标（ :X，:Y）的位置

例 SETXY 60 40

注意如果所用的坐标为负数，那么在负数前面要加一个 ”供 LOGO 识别。

例 SETXY ”－30 20

SETH :H　　海龟转向 :H 度位置，其中海龟头朝上时为 0 度，增加度数时顺时针方向转。

如果想知道海龟当前的位置，用以下命令：

XCOR　　输出海龟当前位置的 X 坐标

YCOR　　输入海龟当前位置的 Y 坐标

HEADING　　输出海龟当前的方向（以度为单位）

所需信息会出现在 RESULT :后面。

例 DRAW
　HEADING

会显示 RESULT :0

再举一例：让海龟从当前位置移到左 20 上 30 的位置

SETXY XCOR－20 YCOR＋30

要注意这一节的命令中，有的用的是任意位置，有的用的是当前位置，可得留神。

让我们来总结一下：

命令	功能
SETX :X	海龟移到横坐标为 :X 的位置
SETY :Y	海龟移到纵坐标为 :Y 的位置
SETXY :X :Y	海龟移到坐标（ :X，:Y）的位置
SETH :H	海龟转向 :H 度位置，其中海龟头朝上时为 0 度，增加度数时顺时针方向转。

XCOR	输出海龟当前位置的 X 坐标
YCOR	输入海龟当前位置的 Y 坐标
HEADING	输出海龟当前的方向（以度为单位）

1.4 进一步了解海龟

还记得如何画正方形、正五边形吧，那样一条边用一个语言行实在麻烦，如果画个正一百边形，岂不要累死。LOGO 提供了一条重复指令，能轻而易举地实现这些。

REPEAT :N〔一组命令〕

此命令是将中括号内的命令重复执行 :N 次，比如画正方形：

REPEAT 4〔FD 50 RT 90〕

画五边形：

REPEAT 5〔FD 50 RT 72〕

REPEAT 命令允许嵌套（即一层里面又套一层），也就是说一个重复命令的执行内容中允许包含另

外的重复命令。在使用嵌套时，中括号要重复出现，如魔球：

REPEAT 10〔REPEAT 5〔FD 30 RT 72〕RT 36〕

由于 LOGO 的基本绘图命令都是绘直线的，所以曲线必须用直线来逼近，就是采用细分的方法，把曲线看作是由很多很短的直线组成的。如画一个圆，可以看作这个圆由 360 段线段组成，每画一段线段，海龟就转动 1 度，等把 360 段线段画完，海龟正好转完 360 度，命令如下：

REPEAT 360〔FD 1 RT 1〕

当然，为了加快速度，也可以少用一些线段来组成圆，但这时画出来的圆弧精度就不够了，比如

REPEAT 36〔FD 10 RT 10〕

你也可只画一段圆弧，只需改变参数就行了。

如果你画了一个圆，你会发现其实它并不是正圆，这是因为不同显示器产生的点与点之间的横距离和纵向距离不一样，为了弥补这一不足，为满足人们的视觉要求，LOGO 设计了横纵比命令：

.ASPECT :X :X 缺省时横纵比为 1

这回再画圆：

.ASPECT 1.4

LT 90

REPEAT 360〔FD 0.5 RT 1〕

显示的将是一个正圆，你也可以用不同的横纵比产生长的或扁的圆，试试吧！注意横纵比命令中必须有那个点，那个点是命令的一部分。

1.5 海龟的特技

键入 HT 命令，海龟就会隐形，这样画好的图形中就可以不出现海龟；键入 ST，海龟就又会现身。

如果有人让你用学过的命令画一条虚线或一个虚圆，你也许会感到束手无策吧。其实就是叫海龟时而画时而不画，重复多次罢了。

PU 提笔不画

PD 落笔画

如虚线 REPEAT 4〔PD FD 5 PU FD 10〕

如虚圆 REPEAT 90〔PD FD 2 RT 2 PU FD 2 RT 2〕

例 DRAW FD 200

这时，海龟从屏幕上方冲了出去，又从屏幕底部爬了出来，这种现象叫作卷绕，即海龟从屏幕 的哪条边界爬出去了，就会从相反的那条边界爬行回来，这种状态是开机后默认的，也可用 WRAP 命令指定。在这种状态下，如果海龟的坐标超出了范围，就会画出意想不到的图形。由此看出海龟的坐标可以设定为 LOGO 允许的实数范围中的任一实数。一般我们都会控制图形的大小，不让它出界，还可以使用 NOWRAP 命令限制。在这种状态下，海龟不能走出边界，否则会显示 TUTRLE OUT OF ROUND（海龟出界了）。

至此，你已经能灵活地控制海龟为你服务了，何不让家人一起分享你的快乐呢！

1.6 五彩缤纷的世界

前面我们一直在画黑白图形，你肯定觉得不过瘾，那么我给你介绍几条彩色绘图的命令。

PICKPEN :N

LOGO 有 4 支画笔（:N=0～3)，其中 1～3 号笔可以画彩色图形，0 号笔是画背景色。

如选用 3 号笔，则 PICKPEN 3

PC :N :X

设定 :N 号笔画的颜色为 :X，:X 的范围是 0～15，色彩代码表请见附录，:N 的范围是 1～3。

如用 2 号笔画绿线，则 PC 2 10

BG :X

设置背景颜色，颜色值 :X 的范围是 0～15。

BRIGHT :N :X

设定 :N 号笔所画颜色的亮度，当 :N=0 时，是设置背景亮度；当 :X=0 时，为低亮度，:X=1 时为高亮度。

这回你如虎添翼，可以画出五彩缤纷的图案了。但要注意，有时两种颜色的线或点位置接近时可能出现相互干涉而串色，因此要精心安排不同颜色的相对位置。另外，也不要把背景的颜色和画笔的颜色设置成一样的，要不屏幕上的图形与背景混在一起，什么也看不出来了。真正的画家除了要有扎实的基本功，更重要的是要独具匠心。祝你早日成为一名计算机画家。

二 过程及其编辑

2.1 过程入门

在上一章我们讲的是交互式绘图，即每画一次图形就必须把所需的指令键入一次，想要再画一次又要键入一次。很多复杂图形或组合图形的绘制或运算是多次执行一组或多组命令来完成的，这些成组的命令就相当于 BASIC 语言中的程序。我们可以把一组实现特定功能的命令当作一个模块，当 我们画组合图形时，只需将这些模块合理地拼装在一起就行了。在 LOGO 语言中，这样的模块有专门 的名称，叫过程。我们可以把需要的命令组编写成过程一次性键入到内存中，使用时就不必反复键入了，只需键入或调用相应的过程名即可。

想让 LOGO 明白你以下将键入的命令是一个过程，就得先定义一下。定义一个过程的方法如下：

在命令状态下键入

TO 过程名

比如 TO A，这条命令的意思就是通知 LOGO："我下面要键入的一组命令叫作过程 A。"过程名是 由字母开头的若干个字母、数字组成，比如 WORLD、CHINA、C3 等等，请注意过程名不能使用 LOGO 命 令如 FD 等，也不允许有空格、括号或运算符号，如 A+、AB * C、(Ь2 等。过程名一般都定义为顾名思 义的名称，可以方便记忆和调用。

2.2 过程的编辑

假设我们定义一个过程，键入

TO A

回车后屏幕被清除，TO A 出现在屏首，这时处于编辑状态，可将所要定义成过程的那一组命令一行一行地输入，当一个语句行超过屏幕宽度时，会自动地在行尾（屏幕最右端）出现一个!，光标移 到下一行行首，你可以继续把该语句行输完。每输完一个语句行按回车键结束。过程结尾都以 END 结束。

如 TO A

REPEAT 4 〔FD 50 RT 90〕

END

在编辑过程中，难免有输入错误或缺漏，这时可使用下面的控制键进行修改。← → ↑ ↓ 移动光标

CTRL－D	删除从光标位置的字符
CTRL－K	删除从光标位置开始直到本语句行末的所有字符
CTRL－A	光标移至语句行首
CTRL－E	光标移至语句行末
CTRL－B	光标上移一屏或移到屏首（第一屏时）
CTRL－F	光标下移一屏或移到屏末（最后一屏）
CTRL－L	将光标当前位置的内容卷至屏首

ENTER (RETURN) 在当前光标处将语句分为两个语句行。当光标处于语句行首时，按此键可在本语句行和上一语句行之间插入一个空行。

CTRL－C　　确认定义，退出编辑。每次键入或修改完一个过程之后都要键入此键退出编辑状态。

CTRL－G　　废除当前的定义，也就是说刚才键入的和修改的内容作废，保持原有的内容，然后退出过程的编辑状态。那么 CTRL－C 是怎样的操作呢，很简单，键盘左边有一个 CTRL 键，你按住它不放，再按一下 C 键 就行了，这就叫 CTRL－C。退出编辑后，LOGO 自动返回纯文本屏。

2.3 过程的调用和运行

调用和运行过程时只要键入过程名就可以了，比如键入上一节我们定义的那个过程的过程名 A，就会画出一个正方形。

在一个过程内，也可以调用其它已定义的过程（包括调用本身），这叫过程的嵌套调用。

例　TO XYZ

SPLITSCREEN

HT

A

FD　50

END 键入 XYZ，运行结果是先画一个正方形，再画一条线段。在过程 XYZ 中，A 就是嵌套调用的过程。

2.4　带参数的过程调用

象正方形过程一样，有很多过程是公用的，那么如何调用同一个过程而画的正方形大小或其它性质不同呢？这里就要使用过程参数了。定义过程时参数并不给出确定的值，而是在调用时再指定。比如设正方形的边长为过程参数，那么就可以在其它过程调用正方形过程时指定参数值，画出边长不同的正方形。

TO 过程名 进入编辑后，按一下←键使光标回到过程名的后面闪烁，然后键入空格和参数名，回车后参数就设定好了。这样一种定义参数的顺序比较特殊，与其它版本的 LOGO 不太一样，得牢牢记住！参数名由字母开头后面带若干个字母或数字组成，在过程中可以把这些参数名当作变量来引用。LOGO 允许定义多个参数，参数之间用空格隔开。

例　定义一个参数为边长 L 的过程

TO A L

REPEAT 4〔FD :L RT 90〕

END

TO A 进入编辑后，按一下←键使光标回到 A 的后面闪烁，然后键入空格和参数名 L，回车后参数 L 就设定好了。

如要画一个边长为 35 的正方形，可在 CTRL－C 退出定义后键入

A 35

过程也可以将一个运算结果返回给调用者，使得过程如同函数。结果的返回采用 OP 命令实现。

OP :X 将 :X 返回给调用者

例 计算 2x－1 的过程

TO C X

OP 2 * X－1

END

注意：过程执行中如遇到 OP 命令，则返回 OP 后面的值，就不再往下之执行本过程，所以 OP 命令一般设在过程的结尾。

2.5　条件语句

在过程执行中，经常判断某些条件是否成立，再根据判断结果执行下去。这样，就要用到条件语句。

条件语句有两种：

IF 条件表达式 THEN 语句 1 ELSE 语句 2

它的意义是：如果条件表达式成立，则执行语句 1，否则执行语句 2。

例 IF :X>0 THEN FD 50 ELSE BK 50

若 :X>0，则海龟前进 50 步，否则后退 50 步。

另一种条件语句是：

IF 条件表达式 THEN 语句 1

它的意思是：如果条件表达式成立，否则执行语句 1。

例 IF :X>0 THEN FD 50
   BK 50

若 :X>0，则海龟前进 50 步，否则后退 50 步。

条件表达式分为简单条件式和复合条件式。

简单条件式有：

:A= :B

:A> :B

:A< :B

复合条件有：

ALLOF	（条件 1）（条件 2）	两个条件都成立时复合条件才成立
ANYOF	（条件 1）（条件 2）	两个条件中只要有一个成立则复合条件成立
NOT	条件	只有条件不成立时复合条件成立

经过条件判断后，经常要执行如下指令：

复合条件的用法如下：

如 MAKE "A 5
   IF ANYOF (:A=5) (:A=4) THEN PR "A=5

意思是先给变量 :A 赋值为 5，然后判断如果 :A=5 或者 :A=4 则显示 A=5

STOP	中止本过程的执行
OP :X	中止本过程的执行，将 :X 返回给调用者
TOPLEVEL	中止本过程的执行，回到执行状态

GO "目标行号：　　跳转到目标号指定的语句行执行。(**注意标号后面一定带冒号，行号可由字母或数字组成，但不能是 LOGO 的命令。还要注意行号后面一定要带有冒号，这个冒号是供 LOGO 识别哪个是行号用的。这个冒号和我们上面提到的象 :X、:M 那样的变量中的冒号位置不一样，它紧跟着写在行号的后面，行号与冒号之间不空格，冒号与后面的命令之间要空格。**)

例 TO A
   10: DRAW
   20: HT
   30: FD 1 RT 1
   GO "30
   END

例 TO A
   ONE: DRAW

```
TWO: HT
LOOP1: FD 1 RT 1
GO "LOOP1
END
```

条件的判断也可用测试命令来实现：

TEST 条件	测试条件是否成立，这里的条件可以是简单条件，也可以是复合条件
IFT 执行语句 1	当被测试的条件成立时执行语句 1
IFF 执行语句 2	当被测试的条件不成立时执行语句 2

经过 TEST 语句后，IFT 与 IFF 可以都出现，或者都不出现，或者出现其中任一个，且在同一过程中，可以 TEST 多次，也可在一次 TEST 后，多次出现 IFT 或 IFF，只是每次的 IFT 或 IFF 都采用最近执行的 TEST 的结果。TEST、IFT、IFF 这些语句之间可以插入多个语句行。

例 TEST :X>0

```
IFT FD 50      如果 :X 大于 0 海龟前进 50 步
IFF BK 50      如果 :X 不大于 0 海龟后退 50 步
IFT BK 100     如果 :X 大于 0 海龟再前进 100 步
```

还有些命令或许对你有用：

POTS	显示已定义的几个过程名
PO 过程名	显示指名的过程清单
POALL	显示所有过程的清单
ER 过程名	删除指名的过程
ERALL	删除所有的过程

2.6 递归

过程可以相互调用，也可以自己调用自己。自己调用自己的过程，称为递归。递归对于编制模块化的高水平程序十分有用。它不仅能使复杂问题简单化，还可以使程序有良好的结构。

例

```
TO ARCH
REPEAT 36 [FD 1 RT 5]
RT 180
ARCH
END
```

因为是自己调用自己，所以当你执行 ARCH 时，会发现它停不下来，这时可用 STOP 键中止正在循环执行的过程。

在递归过程中加入条件语句，就能控制执行的路线，也可判断是否应停止本过程的执行，这称为有条件递归。

例

```
TO A I
IF :I=0 THEN STOP
FD 20 RT 21
A :I-1
END
```

加入条件判断后，这个过程运行的次数就由参数 I 决定了。

如 A 100，就是令过程 A 运行 100 次。

参数递归也是递归中重要的一类。

有人说，使用 LOGO 最有趣的莫过于绘制各种美丽的图形。象蝴蝶对称的翅膀，蜗牛壳上的旋涡线，都可以用递归来实现。深刻体会递归思想，勤于思考会生发无尽的创造。

例　旋涡线

```
TO SQUIRAL SIDE ANGLE
FD :SIDE
RT :ANGLE
SQUIRAL :SIDE+2 :ANGLE
END
```

退出定义后，你可以试试以下几组参数。

```
SQUIRAL 0 89
SQUIRAL 0 90
SQUIRAL 0 91
SQUIRAL 0 72
SQUIRAL 0 73
SQUIRAL 0 118
SQUIRAL 0 119
SQUIRAL 0 143
SQUIRAL 0 144
```

例　海星

```
TO INSPI SIDE ANGLE
HT
FD :SIDE
RT :ANGLE
INSPI :SIDE :ANGLE
END
```

退出定义后，你可以试试以下几组参数。

```
INSPI 12 3
INSPI 10 5
INSPI 15 1
INSPI 15 2
```

其它的组合你自己试试吧！

在这一章的结尾，奉献给大家一棵复杂的递归树。

```
TO TREE SIZE C
IF :C=0 THEN STOP
LT 30
FD :SIZE * 2
TREE :SIZE :C-1
BK :SIZE * 2
```

```
RT 60
FD :SIZE
TREE :SIZE :C-1
BK :SIZE
LT 30
END
```

退出定义后，键入

```
DRAW
HT
PU
SETY "-10
PD
TREE 15 5
```

三 LOGO 的运算功能

3.1 LOGO 的数

(1) 整数

不带小数点的数为整数，LOGO 能处理的整数范围是－2147483647～＋214783647，如果超出这个范围，就把它转化为实数。

(2) 实数

带小数点的数为实数，LOGO 能处理的实数范围是 ±10－38～ ±10＋38，如果超出这个范围，就会因溢出而错误。LOGO 的运算结果只能输出 10 位（包括小数），如果超出这个界限，将采用科学记数法表示，即 10 的几次方用“E”后带次方数表示，如 4×10⁻ 13 表示为 4E－13。

注意，如 10.0 虽然等于 10，但 10.0 是实数，10 是整数。

例如

```
PR 4/7
PR 4000000000/7
PR 40000000000/7
PR 4/7000000000
```

3.2 LOGO 的运算

(1) 四则运算

在 LOGO 中，加、减、乘、除分别表示为＋、－、*、/，并允许使用一层或多层圆括号改变运算优先度。

例 数学式 3× 〔(2－1) ÷ (4－3)〕

LOGO 表达式 3 * ((2－1) / (4－3))

(2) LOGO 的变量

在 LOGO 中，以英文字母开头带有若干个数字或字母为名称并在执行过程中值可以改变的量叫变量，比如 NUMBER、A12、T 等。

变量的值由赋值语句来定义。

MAKE "变量名 数学表达式

例 MAKE "A 30.5 即令 A＝30.5

注意，赋值语句中变量名前必须有一个双引号，当变量被引用时，变量名的前面必须带有冒号。

例　MAKE "N 15

　　FD :N

运行结果是海龟向前走 15 步。

变量赋值后，还可以用赋值语句再次赋值。

例 MAKE "N N＋5

这是把 N＋5 的值再赋给 N。

(3) 函数

LOGO 带有相当丰富的标准函数。所谓标准函数，就是应用广泛，使用方法统一的函数，常见的如三角函数、开方、随机函数等。别着急，下面将举例一一介绍 LOGO 的标准函数。

SIN :X 求正弦函数值（ :X 以度为单价）

例　SIN 45

　　RESULT：.707106781

COS :X　　　求余弦函数值（ :X 以度为单位）

例　COS 90

　　RESULT：0

ATAN :A :B 求余弦值为 :A，正弦值为 :B 的角的度数，即反正切值。

例　ATAN 0 1

　　RESULT：0

SQRT :X 求平方根（:X＞＝0）

例　SQRT 9

　　RESULT：3

ROUND :X 四舍五入取整函数

例　ROUND 2.7

　　RESULT：3

　　ROUND "－2.7

　　RESULT：－3

INTEGER :X 截尾取整函数

例　INTEGER 2.7

　　RESULT：2

　　INTEGER "－2.7

　　RESULT：－3

QUOTIENT :X :Y 商取整函数，把 X/Y 的商取整

例　QUOTIENT 9 2

　　RESULT：4

REMAINDER :X :Y 余数取整函数，把 X/Y 的余数取整

例　REMAINDER 4.6 2

　　RESULT：1

RANDOM :N 随机函数，产生 0 至 N 但不包括 N 的正整数

例　RAMDOM 100

ASCII "字符　　求字符的 ASCII 码

例　ASCII "A

　　RESULT：65

CHAR :N　　　求 ASCII 码为 :N 的字符

例　CHAR 65

　　RESULT :A

函数家族中除了标准函数外，还有一类叫自定义函数，在 LOGO 中我们可以用定义过程的方法自己定义一些常用的函数。

例　定义一个正切函数

```
TO TAN X
OP (SIN :X) / (COS :X)
END
```

退出定义后，我们就创造了一个正切函数。

```
TAN 0
RESULT：0
```

例　定义一个平方函数

```
TO ^ X
OP :X * :X
```

退出定义后，我们就又创造了一个平方函数。

```
END
^ 3
RESULT：9
```

学过这些函数以后，再配合过程的运用，你就能进行多种运算了，还能画很多种函数图形呢！

四　字表处理

4.1 LOGO 的字

由 " 开头并以空格结束的一串字符，叫作一个 LOGO 字。这些字符可以是字母、数字和除了中括号以外的所有其它符号，也可以用双引号后马上用空格结束而成为一个空字，如 "，字中的每一个字符称为字的元素。

例如 "YUXING、1994、"3 * 5，它们都是一个字。值得注意的是，象 1994 这样纯粹由数字组成的字可以不用 "开头。

下面介绍字的运算命令：

FIRST "字	取出字的首元素成为一个单字符的字
如 FIRST "XYZ	的结果为 "X
LAST "字	取出字的末元素成为一个单字符的字
如 LAST "XYZ	的结果为 "Z
BF "字	舍去字的首元素成为由余下字符的字组成的新字

如 BF "XYZ　　　　　　的结果为 "YZ

BL "字　　　　　　　舍去字的末元素成为由余下字符的字组成的新字

如 BL "XYZ　　　　　　的结果为"XY

WORD "字 1 "字 2　　　把字 1 和字 2 连接起来形成新字

如 WORD "YU "XING　的结果为 "YUXING

4.2 LOGO 的表

由一对〔 〕括起的一些字，叫作一个表。表内的字可以不用 "开头，字与字之间用空格隔开，如〔YU 3 * 5 XING〕。表内的每一个字叫作表的元素，一个表也可以作为另一个表的一个元素，如〔3 * 5 〔YU XING〕〕。表内元素的个数叫表长，表内没有元素叫空表。

下面介绍表的运算命令：

FIRST〔表〕　　取出表的首元素

如 FIRST〔1994 YU XING〕的结果为 "1994

LAST〔表〕　　取出表的末元素

如 LAST〔1994 YU XING〕的结果为 "YU XING

BF〔表〕　　　舍去表的首元素

如 BF〔1994 YU XING〕的结果为 "YU XING

BL〔表〕　　　舍去表的末元素

如 BL〔1994 YU XING〕的结果为 "1994 YU

FPUT 元素〔表〕将元素加入表头

如 FPUT 1994〔YU XING〕的结果为〔1994 YU XING〕

LPUT 元素〔表〕将元素加入表尾

如 LPUT "XING〔1991 YU〕的结果为〔1994 YU XING〕

LIST 元素 1 元素 2　将两个元素组成一个表。

如 LIST 1994〔YU XING〕的结果为〔1994〔YU XING〕〕

SE 元素 1 元素 2 将两个元素组成一个表，元素如为表，则自动去掉中括号，它的元素分别成为新表的元素。

如 SE 1994〔YU XING〕的结果为〔1994 YU XING〕

有了这些字表处理命令，就可以很方便地把数字、字母和符号如你所愿地拼接在一起，并对它们进行多样的操作。

五　输入与输出

5.1 输入

有时过程在执行时需要操作者动态地输入一些信息，用来改变变量的值或过程的运行状态。

RC　等待输入一个字符

RQ　等待输入一行字符

例　如果输入 D 则海龟向前 10 步走

```
TO  COMMAND
MAKE "COM RC
IF :COM=D THEN   FD   10
END
```

其实我们前面说过的赋值命令也是输入命令。

5.2 检测

那么，程序又如何判断操作者是否已输入信息是否符合我们所需的类型呢？LOGO 为此设计了一系列检测命令。

RC？　检测是否已键入字符

THING？"A　检测 "A 是不是变量

NUMBER？:X　检测 :X 是不是数字

WORD？:X　检测 :X 是不是字

LIST？:X　检测 :X 是不是表

TS　检测海龟当前的状态

5.3 输出

输出命令在任何计算机语言中都是非常重要的，它可以使操作者了解程序完成了什么工作，正在做什么工作，甚至将要做什么。

PR 表达式 显示命令。可以显示数字、字母、符号、字和表，如果表达式是数学表达式，那么它可以显示计算后的结果。

例 PR "A PR 15 PR〔YU XING〕

PR 15 * 2

PRINT1 表达式紧凑显示。你可以比较一下以下两个语句行的运行结果有何不同。

PR "AAA PR "BBB

PRINT1 "AAA PR "BBB

CURSOR :X :Y 将光标定位在（:X，:Y）处（:X＝2～20 :Y＝3～24)，这样可实现图文并茂。

例 SPLITSCREEN

FD 100

CURSOR 15 10

PR "GAME－OVER

THING "A　输出变量 "A 的值

例 MAKE "B 15

THING "B

5.4 跟踪

当你运行一个由很多语句行组成的过程时，往往想知道程序已运行到哪一句了，这时就需要下面这条命令了。

TRACE　这条命令将逐行跟踪显示运行的那条语句行的行号。

NOTRACE　停止跟踪

如果想保留你键入的过程，可用SAVE "文件名 存入磁带，别忘了按操作手册连接录音机，调出文件时用 READ "文件名，可参阅《入门手册》中有关录音机存调程序的使用说明。

如果你可能用到打印机的话，还可以用命令 CTRL－P 连接打印机打印过程清单，或用 OUT-PRN 命令打印绘制的图形。

错误信息表

1. PROCEURE NO FOUND	未发现有此过程
2. VALURE NO FOUND	无此变量
3. MISS)	遗漏右括号
4. FUNCTION NEED MORE INPUT	参数不够
5. FUNCTION NOT LIKE IT AS INPUT	参数错误
6. NO ENOUGH STORAGE	存储空间不够
7. YOU NO SAY WATH TO DO WITH	不明错误
8. FUNCTION HAVE NO RETURN	函数无返回值
9. FUNCTION CAN ONLY BE USED IN A PROCEDURE	函数只能用于过程中
10. VALURE NAME TOO SHORT	变量名过短
11. ERROR IN PROCEDURE DEFINE	过程定义出错
12. WORLD OR LIST EMPTY	字或表已空
13. FLOAT DVERFLOW ERROR	浮点计算数值范围溢出
14. FUNCTION DIDN'T OUTPUT	函数无输出
15. READ ERROR	读磁带出错

命令一览表

一 绘图命令

FD :N	前进 :N 步
BK :N	后退 :N 步
RT :M	右转 :M 度
LT :M	左转 :M 度
HOME	海龟回到母位，但不清屏
CS	清屏，但海龟不回到母位
DRAW	进入作图，海龟回到母位，清屏
ND	退出作图，图文全清
TEXTSCRENN 或 F1	进入纯文本屏
CLEANTEXT	清除纯文本屏
SPLITSCREEN 或 F2	进入图文混合屏
FULLSCREEN 或 F3	进入纯图形屏
GOODBYE	删除一切，返回初始状态

SETXY :X :Y　　海龟移到（:X，:Y）位置

SETX :X　　海龟移到横坐标为 :X 的位置

SETY :Y　　海龟移到纵坐标为 :Y 的位置

SETH :H　　海龟转向 :H 度位置，其中海龟头朝上时为 0 度，增加度数时顺时针方向转。

XCOR　　输出海龟当前位置的 X 坐标

YCOR　　输入海龟当前位置的 Y 坐标

HEADING　　输出海龟当前的方向（以度为单位）

REPEAT :N〔一组命令〕将中括号内的命令重复执行 :N 次

.ASPECT :X 设置横纵比，:X 缺省时为 1

PICKPEN :N　选择 4 支画笔（:N=0～3)，其中 1～3 号笔可以画彩色。

PC :N :X　设定 :N 号笔画的颜色为 :X，:X 的范围是 0～15，色彩代码表请见附录，:N 的范围是 1～3。

BG :X　　设置背景颜色，即 0 号笔的颜色，:X 的范围是 0～15。BRIGHT :N :X　设定 :N 号笔所画颜色的亮度，当 :N=0 时，是设置背景亮度；当 :X=0 时，为低亮度，:X=1 时为高亮度

二 过程命令

TO 过程名	定义一个过程
END	过程结尾
← → ↑ ↓	移动光标
CTRL－D	删除从光标位置的字符
CTRL－K	删除从光标位置开始直到本语句行末的所有字符
CTRL－A	光标移至语句行首
CTRL－E	光标移至语句行末
CTRL－B	光标上移一屏或移到屏首（第一屏时）
CTRL－F	光标下移一屏或移到屏末（最后一屏）
CTRL－L	将光标当前位置的内容卷至屏首
ENTER（RETURN）	在当前光标处将语句分为两个语句行
CTRL－C	确认定义，退出编辑
CTRL－G	废除定义，退出编辑
OP	将运算结果返回给调用者

IF 条件表达式 THEN 语句 1 ELSE 语句 2 如果条件表达式成立，则执行语句 1，否则执行语句 2。如果没有 ELSE 语句 2，则执行下一语句行。

ALLOF	（条件 1）（条件 2）	两个条件中只有都成立时复合条件才成立
ANYOF	（条件 1）（条件 2）	两个条件中只要有一个成立则复合条件成立
NOT	条件	只有条件不成立时复合条件成立

STOP	中止本过程的执行
OP :X	中止本过程的执行，将 :X 返回给调用者
TOPLEVEL	中止本过程的执行，回到执行状态

GO "目标行号：　跳转到目标号指定的语句行执行。(注意标号后面一定带冒号，行号可由字母或数字组成，不能是命令)。

TEST 条件	测试条件是否成立
IFT 执行语句 1	当被测试的条件成立时执行语句 1
IFF 执行语句 2	当被测试的条件不成立时执行语句 2
POTS	显示已定义的几个过程名
PO 过程名	显示指名的过程清单
POALL	显示所有过程的清单
ER 过程名	删除指名的过程
ERALL	删除所有的过程

三　运算命令

MAKE ”变量名	数学表达式　　赋值
SIN :X	求正弦函数值（:X 以度为单价）
COS :X	求余弦函数值（:X 以度为单位）
ATAN :A :B	求余弦值为 :A，正弦值为：B 的角的度数，即反正切值。
SQRT :X	求平方根（:X>=0）
ROUND :X	四舍五入取整函数
INTEGER :X	截尾取整函数
QUOTIENT :X :Y	商取整函数，把 X/Y 的商取整
REMAINDER :X :Y	余数取整函数，把 X/Y 的余数取整
RANDOM :N	随机函数，产生 0 至 N 但不包括 N 的正整数
ASCII ”字符	求字符的 ASCII 码
CHAR :N	求 ASCII 码为：N 的字符

四　字表处理命令

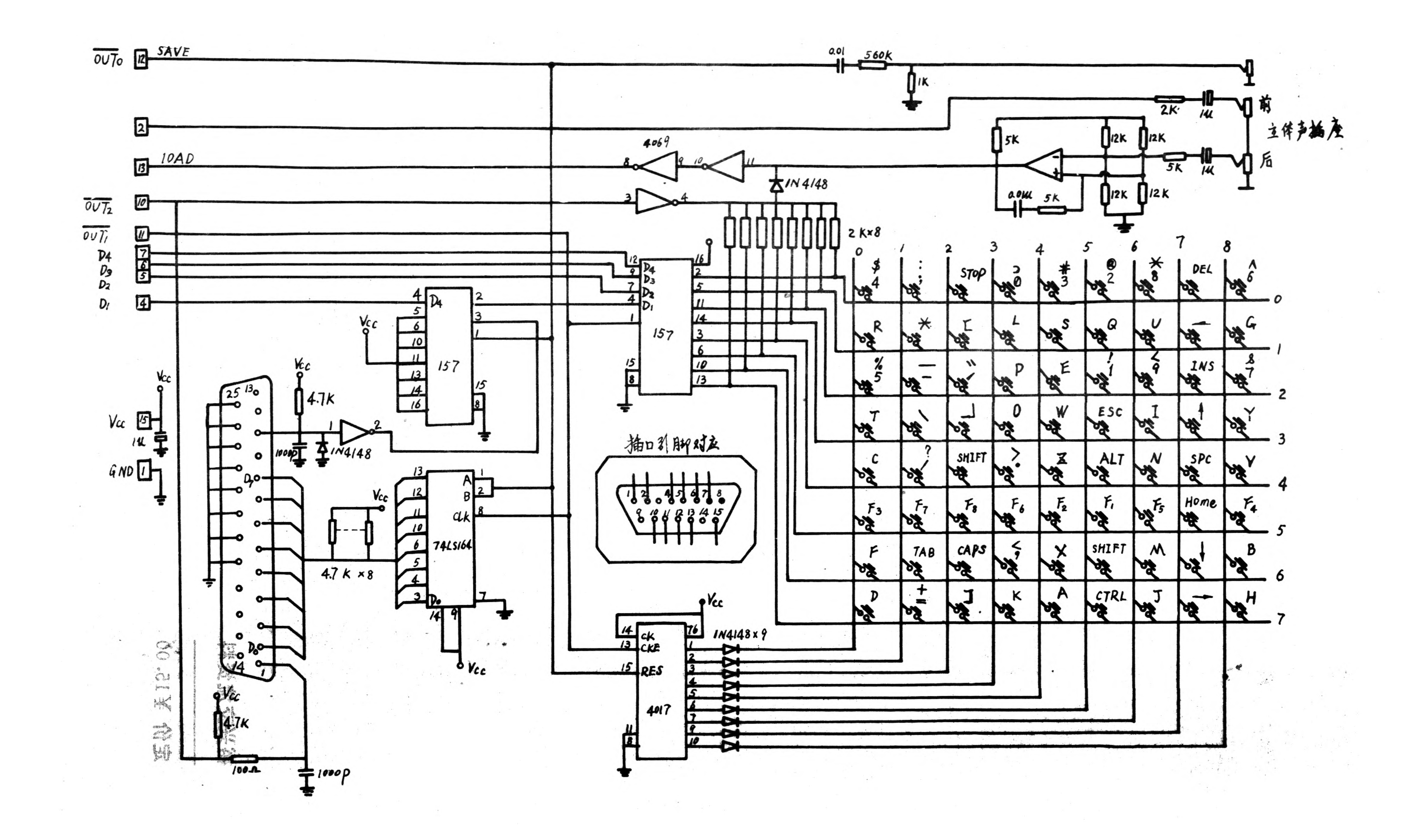
FIRST ”字	取出字的首元素成为一个单字符的字
LAST ”字	取出字的末元素成为一个单字符的字
BF ”字	舍去字的首元素成为由余下字符的字组成的新字
BL ”字	舍去字的末元素成为由余下字符的字组成的新字
WORD ”字 1 ”字 2	把字 1 和字 2 连接起来形成新字
FIRST〔表〕	取出表的首元素
LAST〔表〕	取出表的末元素
BF〔表〕	舍去表的首元素
BL〔表〕	舍去表的末元素
FPUT 元素〔表〕	将元素加入表头
LPUT 元素〔表〕	将元素加入表尾
LIST 元素 1 元素 2	将两个元素组成一个表。
SE 元素 1 元素 2	将两个元素组成一个表，元素如为表，则自动去掉中括号，它的元素分别成为新表的元素。

五 输入与输出命令

RC	等待输入一个字符
RQ	等待输入一行字符
RC?	检测是否已键入字符
THING? "A	检测 "A 是不是变量
NUMBER? :X	检测 :X 是不是数字
WORD? :X	检测 :X 是不是字
LIST? :X	检测 :X 是不是表
TS	检测海龟当前的状态
PR 表达式	显示命令
PRINT1 表达式	紧凑显示
CURSOR :X :Y	将光标定位在屏幕的（:X，:Y）处（:X=2～20 :Y=3～24）
THING "A	输出变量 "A 的值
TRACE	逐行跟踪显示运行的那条语句行的行号
NOTRACE	停止跟踪
SAVE "文件名	将文件存入磁带
READ "文件名	将文件读入内存
CTRL-P	连接打印机打印过程清单
OUTPRN	打印绘制的图形

色 码 表

0	白	1	蓝	2	蓝紫	3	紫	4	紫红	5	粉红	6	红	7	橙
8	黄	9	黄绿	10	绿	11	绿	12	浅蓝	13	灰	14	黑	15	透明

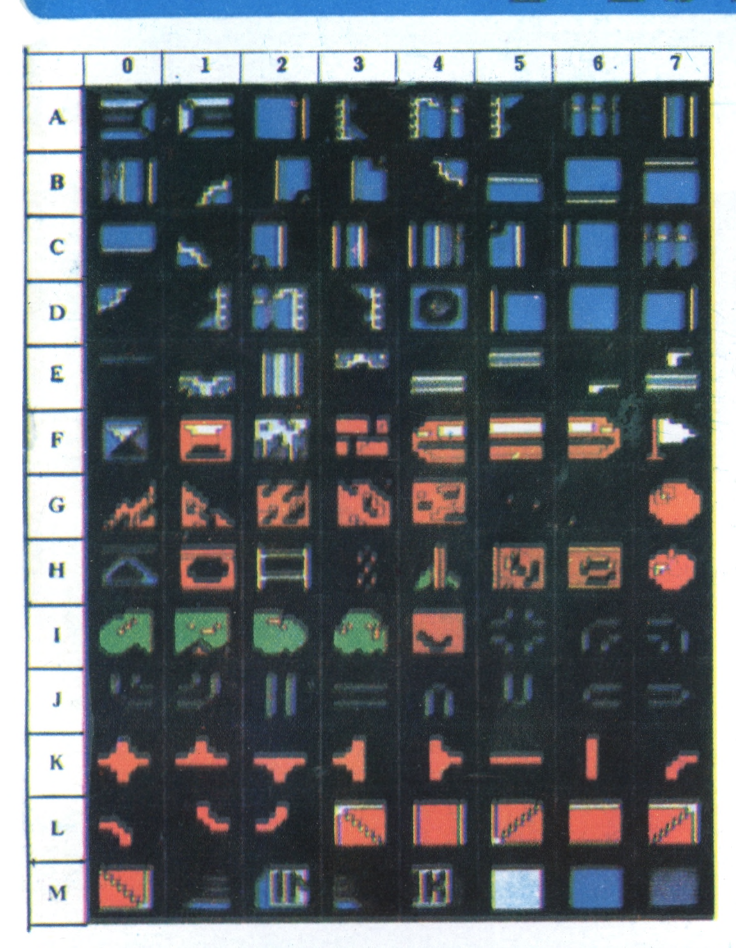


SAVE
10AD
4069
IN4148
2K×8
0.01
560K
1K
2K
5K
12K
前
后
立体声插座
157
74LS164
4017
4.7K
4.7K×8
1000P
100Ω
IN4148×9
插口引脚对应
Vcc
GND
DEL
INS
SPC
Home
ESC
ALT
SHIFT
CTRL
STOP
CAPS
TAB

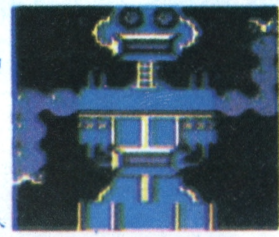
裕兴交流资料

定价 ¥15.00

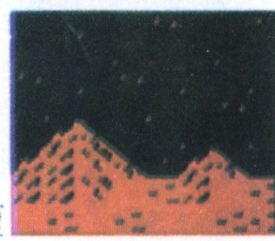
图B 图形表



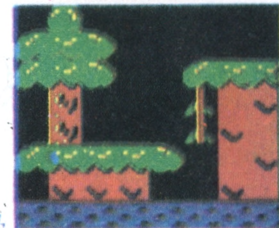
利用BG绘图程序的符号可描绘各种的图形。右图为其例子。



机器人

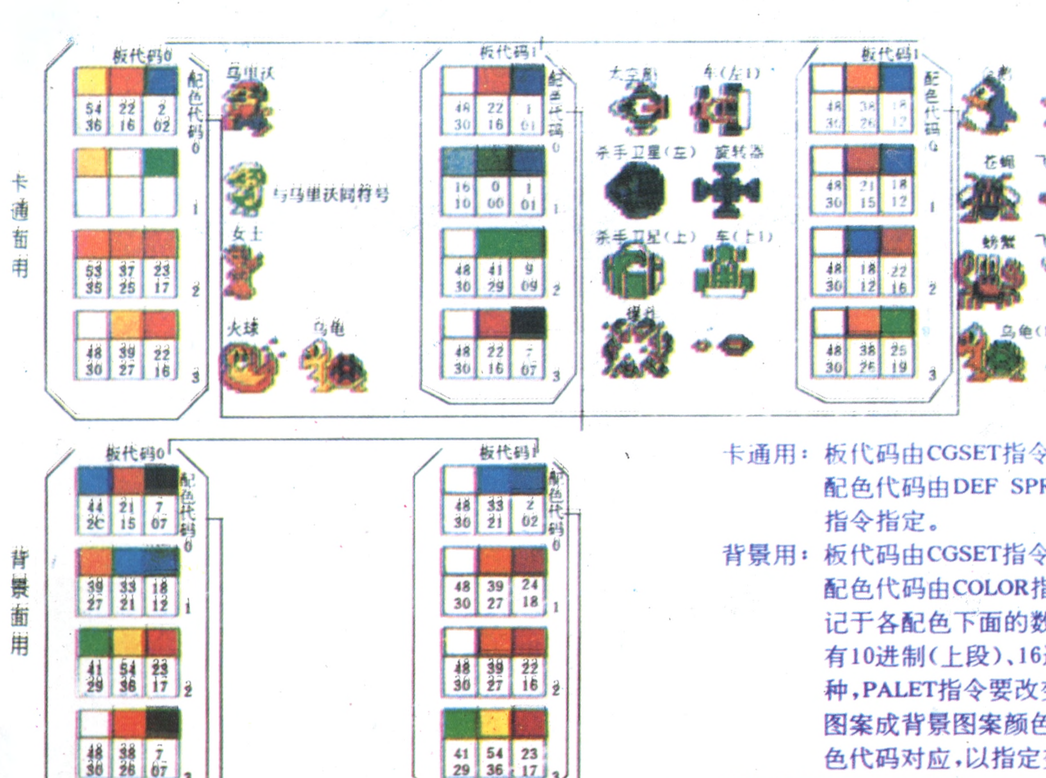


山和星星



树和岛

图C 色彩图表



卡通用：板代码由CGSET指令指定；
配色代码由DEF SPRITE、DEF MOV指令指定。

背景用：板代码由CGSET指令指定；
配色代码由COLOR指令指定。
记于各配色下面的数字(颜色代码)有10进制(上段)、16进制(下段)二种，PALET指令要改变其卡通本身的图案成背景图案颜色时，请与这些颜色代码对应，以指定变更颜色。

颜色代码用PALET指定

图A　卡通图案



各图形四角的号码为10进制，在SPRITE指令中CHR$(n)中使用。


家用电脑学习机操作手册